



Pro gradu-avhandling
Meteorologi

Optimering av parametrar i en idealiserad atmosfärmodell

Madeleine Ekblom

25.10.2017

Handledare: Professor Heikki Järvinen

Granskare: Docent Jouni Räisänen
Professor Heikki Järvinen

HELSINGFORS UNIVERSITET
INSTITUTIONEN FÖR FYSIK

PB 64 (Gustaf Hällströms gata 2)
00014 Helsingfors universitet



HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

MATEMAATTIS-LUONNONTIEELLINEN TIEDEKUNTA
MATEMATISK-NATURVETENSKAPLIGA FAKULTETEN
FACULTY OF SCIENCE

Tiedekunta – Fakultet – Faculty Matematisk-naturvetenskapliga		Koulutusohjelma – Utbildningsprogram – Degree programme Utbildningsprogrammet för de fysikaliska vetenskaperna	
Tekijä – Författare – Author Madeleine Eklom			
Työn nimi – Arbetets titel – Title Optimering av parametrar i en idealiserad atmosfärmodell			
Työn laji – Arbetets art – Level Pro Gradu-avhandling	Aika – Datum – Month and year Oktober 2017	Sivumäärä – Sidoantal – Number of pages 38 sidor	
Tiivistelmä – Referat – Abstract <p>Numeriska väderprognosmodeller (NWP, numerical weather prediction) innehåller variabler, såsom vindvektorer, temperatur och tryck, samt modellparametrar. Modellparametrar är en oundviklig del av en NWP då de behövs i parametriseringsscheman av småskaliga fysikaliska processer. Modellparametrarna har betydelse för modellens förmåga att prognostisera vädret och det är därav av stor vikt att parametrarna har optimala värden.</p> <p>Värdet på parametrarna bestäms manuellt med hjälp av observationer och expertkunskap, vilket är en arbetsdryg process. För att minska på arbetsbördan och frigöra resurser för annat är det möjligt att använda sig av algoritmer för att bestämma de optimala värdena på modellparametrarna. I den här studien undersöker vi en statistisk algoritm som används i ett ensembleprognossystem. Algoritmen grundar sig på att uppdatera och förbättra parametervärdena baserat på observationer och en kostnadsfunktion.</p> <p>Tidigare resultat har visat att algoritmen ger goda resultat med en ensemble på 50 medlemmar. I den här studien undersöks huruvida det är möjligt att minska ensemblestorleken vid optimering av två parametrar och samtidigt uppnå goda resultat. I experimenten används Lorenz96-modellen för att emulera NWP och ensemblestorleken varieras mellan två och 50. Resultaten tyder på att en ensemblestorlek på fyra ger goda resultat då utgångsläget är optimalt och vid mindre optimala förhållanden krävs flera ensemblemedlemmar; i det här fallet tio.</p> <p>Baserat på de här experimenten kan vi dra slutsatsen att algoritmen inte begränsas av ensemblestorleken utan snarare av modellen och utgångsläget vid parameteroptimering. Ju bättre utgångsläget är, desto mindre ensemble är möjligt att använda. Experimenten indikerar på att motsvarande resultat är att vänta för NWP.</p>			
Avainsanat – Nyckelord – Keywords Lorenz96, parameteroptimering, ensembleprognos			
Säilytyspaikka – Förvaringställe – Where deposited			
Muita tietoja – Övriga uppgifter – Additional information			

Innehåll

1	Introduktion	1
2	Teori	6
2.1	Lorenz96-modell	6
2.2	Parameteroptimering	10
2.2.1	Bakgrund	11
2.2.2	Algoritm	12
3	Numeriska experiment	16
3.1	Numeriska modeller	16
3.2	Experimentupplägg	18
3.2.1	Exempel - uträkningar	21
4	Resultat	24
4.1	Experiment med liten ensemble	24
4.1.1	Konvergenstest	24
4.1.2	Experiment 1 - mer optimal a priori distribution	25
4.1.3	Experiment 2 - mindre optimal a priori distribution	28
4.2	Verifiering av hypotesen	29
5	Sammanfattning	34
	Litteraturförteckning	37
	Bilagor	39
A	Lista över parametrar och variabler	39

Kapitel 1

Introduktion

Numeriska väderprognosmodeller grundar sig på Navier-Stokes ekvationer för flödesdynamik i atmosfären. Ekvationssystemet består av flera sammanbundna ekvationer, vilket gör ekvationssystemet mycket komplicerat att lösa och det är oklart huruvida det ens existerar en analytisk lösning till systemet. Problemet är ett av millenniumpreisproblemen utfärdade av Clay Mathematics Institute år 2000. En utförlig beskrivning av problemet finns på:

<http://www.claymath.org/millennium-problems/navier-stokes-equation>.

Historien för moderna väderprognoser sträcker sig tillbaka till 1900-talets början (Lynch, 2008). De första stegen som så småningom skulle leda till genombrott var Cleveland Abbes förslag att använda matematiska metoder för att göra prognoser. Bjerknes gjorde senare flera framsteg då han föreslog sju variabler vilka han ansåg viktiga för väderprognoser: tryck, temperatur, densitet, fuktighet samt vindvektorer i x-, y-, och z-riktning. Därutöver föreslog han ekvationer vilka kan användas för att göra prognoser för de ovannämnda variablerna. Bjerknes kunde varken lösa ekvationerna numeriskt eller analytiskt utan han löste dem med grafiska metoder.

Richardson skulle senare komma att föreslå numeriska lösningsmetoder för ekvationerna. Han använde sig av finita differensmetoden för att lösa de ekvationer som tidigare lyfts fram av Abbe och Bjerknes. Richardsons första uträkningar med trycktendenser visar sig dock vara orealistiska, vilket till stor del berodde på att han hade begränsade resurser att hantera både observationerna och beräkningarna. I sin bok *Weather Prediction by Numerical Process* (1922) presenterar han sina idéer i vilka moderna väderprognoser har sin grund.

De tidigaste approacherna till väderprognoser gjordes för hand och de var således arbetssamma. Då de första datorerna utvecklades på 1950-talet kom de första operativa väderprognoser att bli möjliga. De första väderprognoserna var

helt deterministiska, vilket senare visade sig vara en begränsande faktor för att kunna göra längre och samtidigt tillförlitliga prognoser. Lorenz (1963, 1969) visade att deterministiska väderprognoser begränsas av det faktum att små fel i initialvärdena för väderprognosmodellerna kan växa sig ofantligt stora under en relativt kort tid. Orsaken till det här är atmosfärens kaotiska karaktär som gör att två till synes lika initialvärden kan ta två olika vändningar. Som lösning på problemet undersökte Lorenz ensembleprognoser, d.v.s. en grupp av prognoser vars initialvärden skiljer sig lite från varandra, för att kunna analysera hur tillförlitlig en prognos är. Därmed satte Lorenz grund för kaosteori och lade grunden för ensembleprognoser. Grunden för ensembleprognoser lades alltså fram på 1960-talet och har under årens lopp förfinats till en process som idag används i mer eller mindre alla väder- och klimatprognoser (Slingo & Palmer, 2011).

Ensembleprognoser i operativa väderprognoser har varit rutin sedan mitten på 1990-talet (Leutbecher & Palmer, 2008; Lewis, 2005). Ensembleprognoser är i grund och botten en form av Monte Carlo-metod, vilken kräver någon form av initialvärden. Skillnaden mellan en traditionell Monte Carlo-metod och den i ensembleprognoser finns i sannolikhetsfunktionen; i en traditionell Monte Carlo-integrering är den känd medan den i en ensembleprognos är mer eller mindre okänd. För att få fram bra väderprognoser krävs det därför en rimlig representation av prognosfelet genom att välja lämpligt stora avvikelser mellan initialvärdena.

Precis som Richardson föreslog i början av 1900-talet så bygger dagens väderprognoser på numeriska lösningar baserade på finita differensmetoden av Navier-Stokes ekvationer (t.ex. (Chorin, 1968)). För att kunna få fram en lösning krävs det ett initialvärde från vilket prognosen sedan kan beräknas. Initialvärdet fås från observationer från exempelvis satelliter och väderstationer. Den finita differensmetoden innebär att både tid och rum delas in i ändliga differenser så att jordklotet indelas i ett så kallat rutsystem. Ju mindre rutor som används, desto mer beräkningsresurser krävs det för att få fram en prognos. På grund av att rutorna inte kan vara hur små som helst så innebär det även att processer vars storleksklass är mindre än en ruta inte kommer att synas i själva prognosmodellen. Av den här orsaken så införs parameteriseringsscheman vilka fångar upp effekten av de här småskaliga fysikaliska processerna, såsom processer relaterad till konvektion och molnbildning (Kain, 2004; Anthes, 1977). De småskaliga processerna påverkar de storskaliga processerna och således även väderprognoserna, vilket gör att modellparametrar är en oundviklig del av en numerisk vädermodell.

Initialvärdesfel är en källa för fel i väderprognoser, delvis löst genom använd-

ningen av ensembleprognoser. Det finns dock alltid utveckling för förbättring genom noggrannare observationer tätare spridd över jorden. Utöver initialvärdesfel så finns det andra felkällor i väderprognosmodeller; modellerna innehåller systematiska fel vilka uppkommer till följd av exempelvis inexakta parametreringsscheman (Palmer, 2001). Modellparametrar är en oundviklig del av väderprognosmodeller eftersom småskaliga fysikaliska processer måste parametreras för att inte deras effekt ska försvinna helt. Värdet på modellparametrarna bestäms utgående från observationer och expertkunskap, vilket är en tidskrävande och arbetsdryg process ifall den görs för hand.

I stället för att bestämma modellparametrarnas värde manuellt så är det möjligt att använda sig av optimeringsalgoritmer. Navon (1998) diskuterar olika metoder för att estimerar parametrar i kontexten meteorologiska och oceanografiska modeller. Navon nämner problemet med att identifiera ett unikt parametervärde och ger exempel på olika metoder som kan användas för att optimera parametervärden: maximum likelihood (ML), total variation regularization (TVR) och extended Kalman filter (EKF). Till sist påminner Navon att parametrar i väderprognosmodeller kan vara beroende av årstid och andra fysiska påverkningar, vilket bör tas i beaktande vid optimering av parametrarna.

I den här studien studeras en algoritm som bygger på att estimerar parametrar i ett ensembleprognossystem: EPPES (*eng.* ensemble prediction and parameter estimation system, fritt översatt till ensembleprognos- och parameteruppskattningssystem) (Järvinen *et al.* , 2012; Laine *et al.* , 2012). Algoritmen bygger på att undersöka hur ett parametervärde eller en parametervektor påverkar en enskild ensemblemedlems förmåga att prognostisera. Genom att variera parametervärdena i en ensembleprognos så att varje enskild ensemblemedlem har unika parametervärden fås ett stort urval av parametrar och deras respektive prognosförmåga. Utgående från deras prognosförmåga uppdateras parametervärdena så att nästa ensembleprognos får nya, bättre parametervärden. Metoden möjliggör uppskattning av parametrar i ett operativt väderprognossystem då själva ensembleprognosen är en fristående enhet i algoritmen och den enda förändring som krävs i ensembleprognossystemet är införandet av varierande parametervärden.

Algoritmen bygger på att sampla parametervärden för en given ensemblestorlek. Tidigare resultat med algoritmen visar att det är möjligt att estimerar ett litet antal parametrar med en ensemble på 50 medlemmar. De första experimenten med algoritmen visar att optimering av två parametrar för en liten modell uppnår goda resultat vid användning av 50 ensemblemedlemmar (Laine *et al.* , 2012). I den studien använder Laine *et al.* (2012) sig av Lorenz96-modellen

(Lorenz, 1996) för att emulera väderprognosmodeller i ensemblesystemet. Lorenz96 är en icke-linjär, dynamisk modell vars dynamik är kaotisk vid lämpliga förhållanden. Senare resultat med mer krävande modeller bekräftar algoritmens förmåga att optimera ett litet antal parametrar med en ensemble på 50 medlemmar. Den första studien med klimatmodellen ECHAM5 med resolution T42L31 visar att algoritmen kan optimera fyra parametrar med 50 ensemblemedlemmar (Ollinaho *et al.*, 2013a). I den andra studien använder Ollinaho *et al.* (2013b) sig av ECMWF IFS-modell med resolution T159L62 för att optimera fyra parametrar med 50 ensemblemedlemmar. Studien visar att EPPEs-algoritmen har förmågan att optimera parametrar så att deras prognosförmåga blir bättre än standardvärdenas.

De tidigare resultaten ger mer att önska och det är värt att ställa frågan huruvida det är möjligt att minska ensemblestorleken och samtidigt få goda resultat. En stor ensemble kräver mer beräkningsresurser och det är därför skäl att överväga hur stor ensemble som behövs för det önskade resultatet. I den här studien vill vi undersöka ifall det är möjligt att minska ensemblestorleken så till vida att dess storlek motsvarar parametrarnas antal. Vid optimering av många parametrar är det viktigt att veta vilka faktorer som begränsar algoritmen samt hur mycket resurser som krävs för att nå ett önskat resultat. Vi utgår från att antalet parametrar och ensemblestorleken bör vara i samma storleksklass för att konvergens skall uppnås. Hypotesen lyder som följer:

Hypotes. *Låt L vara antalet parametrar som önskas optimeras och M vara antalet ensemblemedlemmar. Vid optimering av parametrar med EPPEs så räcker det med att antalet parametrar och antalet ensemblemedlemmar är i samma storleksklass för att konvergens skall uppnås.*

Vi kommer att undersöka hypotesen med Lorenz96-modellen. Modellen kräver minimala datorresurser, vilket möjliggör att experimenten kan utföras med en bärbar dator. En första undersökning med Lorenz96-modellen ger indikationer på hur algoritmen beter sig med en liten ensemble samt hurudana resultat som är att vänta vid användning av en mer krävande modell. Då Lorenz96-modellen inte kräver särskilt mycket resurser så kan vi utföra flera experiment för att besvara hypotesen.

I det här fallet undersöker vi hur många ensemblemedlemmar som krävs för att optimera två parametrar i Lorenz96-modellen. Vi kommer att använda oss av två olika varianter av Lorenz96-modellen, där den ena varianten används för att generera syntetiska observationer och den andra innehåller ett parametrerings-schema för småskaliga processer. Vi vill optimera parametrarna i parametrings-

schemat så att modellen så bra som möjligt motsvarar de syntetiska observationerna. Vi kommer att utföra två experiment, där det första har ett mer optimalt utgångsläge för att optimera parametrarna och det andra ett mindre optimalt utgångsläge. Vi förväntar oss att ett mer optimalt utgångsläge ger ett bättre resultat vid användning av algoritmen. I övrigt skiljer sig de två experimenten inte från varandra.

Kapitel 2 beskriver Lorenz96-modellen och algoritmen för parameteruppskattning i detalj med hjälp av matematiska formler och illustrationer. I kapitel 3 beskrivs upplägget för de numeriska experimenten och ett konkret exempel visar hur algoritmen fungerar. I kapitel 4 presenteras resultaten från de två experimenten och hypotesen diskuteras i relation till resultaten. Till sist sammanfattas studien i kapitel 5.

Kapitel 2

Teori

Det här kapitlet behandlar Lorenz96-modellen och dess olika varianter samt parameteruppskattningsalgoritmen EPPES. Lorenz96-modellen har en dynamik som lämpar sig för att emulera numeriska väderprognosmodeller i undersökningar med ensemblesystem. EPPES är en algoritm som används för att estimerar parametrar i ett ensemblesystem. Algoritmen bygger på bayesiansk statistik och följer en hierarkisk metod för att estimerar modellens parametrar.

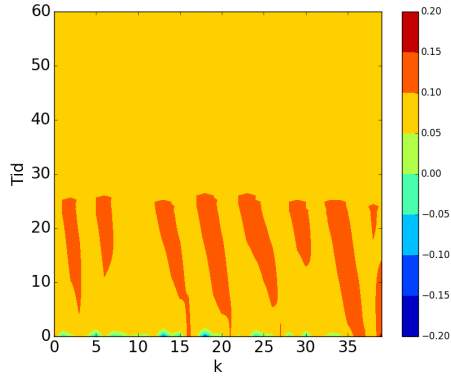
2.1 Lorenz96-modell

Numeriska väderprognosmodeller (NWP, numerical weather prediction) kräver enorma mängder beräkningsresurser och det finns därför ett behov för en mindre krävande modell som kan emulera vädermodellerna i olika undersökningar med ensemblesystem. Lorenz (1996) beskriver en dynamisk modell, Lorenz96-modellen (L96), som kan användas för det här syftet. Modellen består av K variabler X_1, \dots, X_K och en konstant kraft F . Variablernas dynamik bestäms enligt

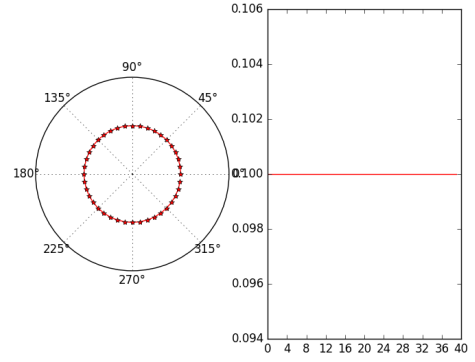
$$\frac{dX_k}{dt} = -X_{k-2}X_{k-1} + X_{k-1}X_{k+1} - x_k + F. \quad (2.1)$$

Systemet är periodiskt, det vill säga $X_{K+1} = X_1$ och $X_0 = X_K$, vilket gör att Lorenz96-modellen kan beskrivas som ett system med en variabel runt en breddgrad. Då motsvarar X en storhet vilken uppmäts eller observeras med $360/K$ graders mellanrum och X_k det uppmätta värdet vid längdgrad $k \cdot 360/K^\circ$.

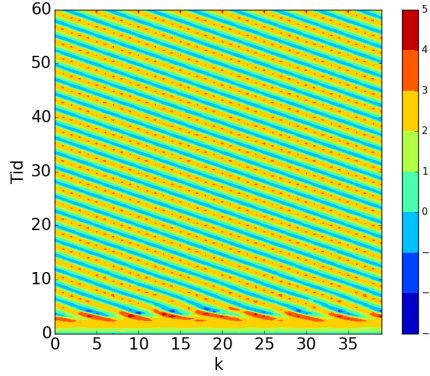
L96-modellens dynamiska karaktär beror på värdet på forceringstermen F . Vid mycket låga värden på forceringstermen kommer systemet att anta konstanta värden och för något större värden uppstår ett periodiskt system. Då antalet variabler är 40 så ger ett forceringsvärde på 10,0 ett kaotiskt L96 system.



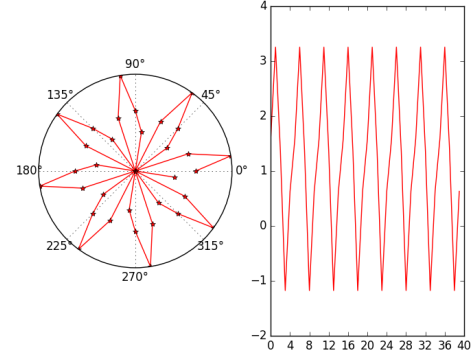
(a) $F=0,1$. Konstant system.



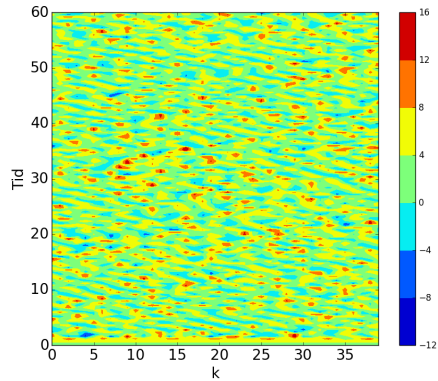
(b) $F=0,1$. X_k konstant.



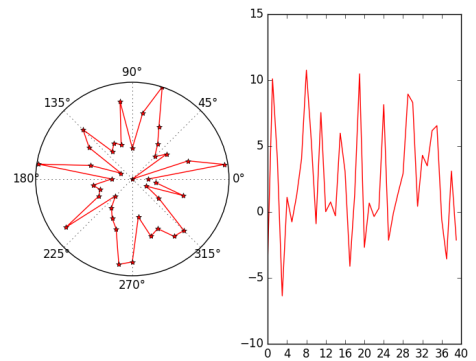
(c) $F=3,0$. Periodiskt system.



(d) $F=3,0$. X_k är periodisk.



(e) $F=10,0$. Kaotiskt system.



(f) $F=10,0$. X_k är kaotisk.

Figur 2.1: Hovmöllerdiagram för L96-modellen med olika värden på forcingtermen (vänster kolumn) samt motsvarande (k, X_k) -diagram för de olika forceringsvärdena (höger kolumn). (k, X_k) -diagrammen är plottade vid en godtycklig tidpunkt i såväl polära som kartesiska koordinater för att förtydliga effekten av värdet på forceringen.

Idén bakom figur 2.1 är baserad på (Karimi & Paul, 2010), i vilken de studerar Lorenz96-modellens kaotiska dynamik. Figur 2.1 ger en bild av hur dynamiken ser ut vid en given tidpunkt och över en längre tidsperiod. Figuren innehåller sex diagram; tre stycken Hovmöller-diagram¹ med olika forceringstermer och tre motsvarande diagram över tillståndsvariabeln X_k som funktion av k vid en god-

¹Hovmöller-diagram visar hur vågor rör sig i tid och rum (Hovmöller, 1949).

tycklig tidpunkt. Från de två översta diagrammen ser vi att ett litet värde på F leder till ett konstant, oföränderligt läge. Alla variabler når värdet på forceringstermen, vilket syns i att Hovmöller-diagrammet når ett konstant värde och (k, X_k) -diagrammet förtydligar att X_k är konstant för alla värden på k . De två mellersta diagrammen visar att systemet antar en periodisk karaktär då värdet för F är något större, i det här fallet är $F=3,0$. Periodiciteten syns i Hovmöller-diagrammet på att det uppstår ränder, vilka visar på att värden återkommer med ett visst intervall. Diagrammet över X_k som funktion av k förtydligar det här genom att visa hur X_k varierar periodiskt. De två nedersta diagrammen visar liksom tidigare ett Hovmöller-diagram och ett (k, X_k) -diagram, i det här fallet ett system med en forceringsterm på $F = 10,0$. Systemet är kaotiskt och det finns ingen antydning på att systemet skulle gå mot ett konstant eller ett periodiskt läge.

Ekvation (2.1) beskriver den enklaste formen av L96. Lorenz (1996) beskriver även en variant av L96 som består av två ekvationer. Den första ekvationen beskriver de så kallade långsamma tillståndsvariablernas, X_k , dynamik och den andra ekvationen beskriver de så kallade snabba tillståndsvariablernas, U_j , dynamik. Dynamiken för de långsamma och de snabba variablerna följer

$$\frac{dX_k}{dt} = -X_{k-2}X_{k-1} + X_{k-1}X_{k+1} - x_k + F + (hc/b) \sum_{j=1}^J U_{j,k} \quad (2.2a)$$

respektive

$$\frac{dU_{j,k}}{dt} = -cbU_{j+1,k}(U_{j+2,k} - U_{j-1,k}) - cU_{j,k} + F_u + (hc/b)X_k, \quad (2.2b)$$

där $k = 1, \dots, K$, $j = 1, \dots, J$, F är en forceringsterm, $(hc/b) \sum_{j=1}^J U_{j,k}$ beskriver de snabba variablernas forcering på de långsamma variablerna och h, c och b är parametrar. Systemet kan som tidigare beskrivas som ett system med K tillståndsvariabler runt en breddgrad vilka forceras på grund av utomstående krafter. De här utomstående krafterna är forceringstermen F samt den kraft som de snabba variablerna $U_{j,k}$ ger upphov till. De snabba variablerna kan även de beskrivas som en storhet som varierar runt en breddgrad. Den kraft som verkar på de snabba variablerna är F_u , vilken ibland utelämnas från beskrivningen av de snabba variablernas dynamik.

Lorenz (1996) jämför de snabba variablerna med småskaliga processer i atmosfären och de långsamma variablerna med mer storskaliga processer. Som exempel ger han en process relaterad till konvektion, där de snabba variablerna

representerar någon storhet associerad med konvektiv skala och de långsamma variablerna statisk instabilitet. Precis som med konvektiva atmosfäriska processer kan de snabba variablernas effekt parametreras.

Wilks (2005) beskriver en parametriserad version av L96-systemet i ekvation (2.2), där de snabba variablernas effekt ersätts med en parametrisering. Den parametriserade versionen kan beskrivas som

$$\frac{dX_k}{dt} = -X_{k-2}X_{k-1} + X_{k-1}X_{k+1} - x_k + F - g(X_k, \theta), \quad (2.3)$$

där $g_k(\theta, t)$ beskriver funktionen för parametriseringen av de snabba variablerna $U_{j,k}$. Parametriseringen kan göras på olika sätt och i den här studien betraktas enbart linjär parametrisering

$$g_k(\theta, t) = \theta_1 + \theta_2 X_k. \quad (2.4)$$

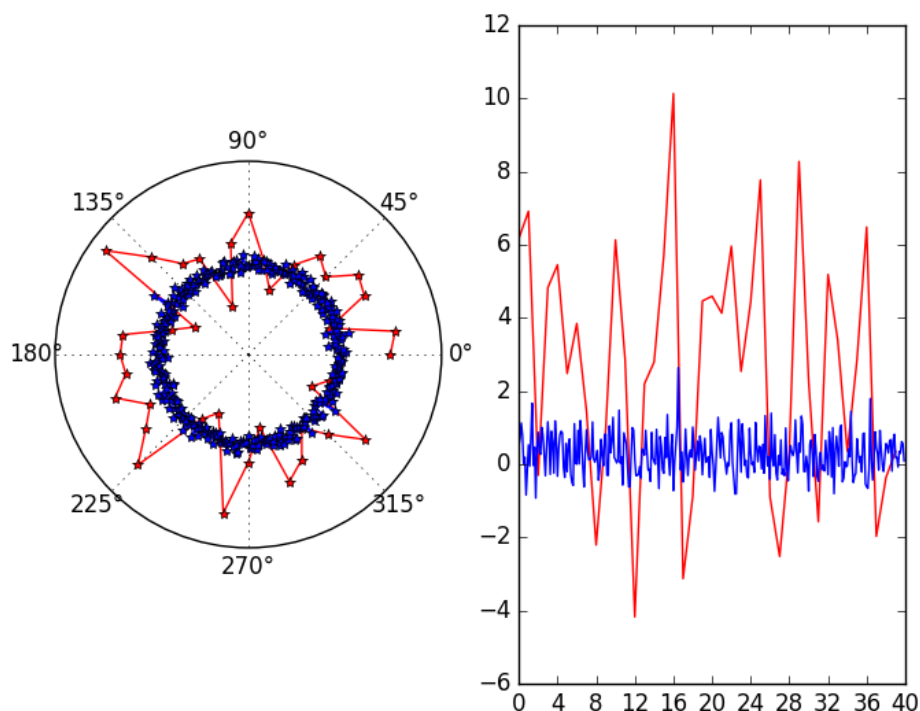
Det finns även varianter av ekvation (2.4) där parameteriseringen av de snabba variablerna även innehåller en stokastisk forcering för att kompensera för osäkerheten som parameteriseringen innebär. I sådana fall beskrivs parameteriseringen så som följer

$$g_k(\theta, t) = \theta_1 + \theta_2 X_k + \eta_k(t),$$

där $\eta_k(t)$ representerar den slumpmässiga forceringen.

Figur 2.2 visar hur de långsamma och de snabba variablerna ter sig runt en breddgrad i enlighet med ekvation (2.2). De långsamma variablerna, representerade av den röda linjen, har större amplitud än de snabba variablerna representerade av den blåa linjen.

Vid lämpliga värden på forceringstermen har L96-modellen en kaotisk dynamik, vilket gör att modellen lämpar sig för att emulera vädermodeller i ensemble-prognoser samt för att undersöka hur olika algoritmer och metoder fungerar med ett kaotiskt system. Den parametriserade versionen av L96 lämpar sig även för experiment med uppskattning av parametrar.



Figur 2.2: Diagram över de snabba, $U_{j,k}$, och de långsamma, X_k , variablerna som funktion av k . Det högra diagrammet visar de verkliga värdena på variablerna medan det vänstra diagrammet visar schematiskt hur variablerna ter sig runt en breddgrad. Den röda linjen representerar de långsamma, storskaliga variablerna och den blåa linjen representerar de snabba, småskaliga variablerna. En parametrisering innebär att effekten från de småskaliga variablerna ersätts med ett parametriseringsschema.

2.2 Parameteroptimering

EPPES (Järvinen *et al.*, 2012), ensemble prediction and parameter estimation system (*ung.* ensembleprognos- och parameteruppskattningssystem), är en algoritm för att uppskatta parametervärden samtidigt som en ensembleprognos körs. Algoritmen grundar sig på en hierarkisk metod för att uppskatta parameterosäkerhet och på en metod kallad viktbaserad sampling (*eng.* importance sampling). Den här sektionen är uppdelad i två delar: den första delen behandlar den statistiska bakgrund som behövs för att förstå EPPES och den andra delen behandlar algoritmen och dess funktion.

2.2.1 Bakgrund

EPPEs grundar sig på metoder som baserar sig på Bayesiansk statistik. Bayesiansk statistik har sin grund i Bayes sats:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)},$$

där $P(A)$ är sannolikheten för händelse A även kallad a priori, $P(B)$ är sannolikheten för händelse B , $P(A|B)$ är sannolikheten för händelse A givet B (även kallad a posteriori) och $P(B|A)$ sannolikheten för B givet A (även kallad trovärdighet). Det är möjligt att beskriva a posteriori som

$$P(A|B) \propto P(B|A)P(A).$$

Alltså, a posteriori sannolikheten är proportionell mot produkten av trovärdigheten och a priori sannolikheten. Den här egenskapen hos a posteriori öppnar möjligheter för olika statistiska metoder. Till näst följer en kort beskrivning av begreppen Markov Chain Monte Carlo-metoder och viktbaserad sampling, vilka är grunden för EPPEs.

Definition 1 (Markov-kedja). *En Markov-kedja är uppbyggd av en kedja av slumpmässiga variabler X_1, \dots, X_t vilka samplas från en distribution $P(X_{t+1}|X_t)$. En stationär distribution är den distribution till vilken Markov-kedjan konvergerar.*

Definition 2 (Monte Carlo-integrering). *Monte Carlo-integrering strävar till att uppskatta $E(f(X))$ genom att sampla värden för X från en sannolikhetsfördelning $\pi(\cdot)$ och sedan uppskatta*

$$E(f(X)) \approx 1/n \sum_{t=1}^n f(X_t).$$

Definition 3 (Markov Chain Monte Carlo, MCMC). *En MCMC-metod är en Monte Carlo-metod, där sannolikhetsfördelningen $\pi(\cdot)$ är den stationära distributionen i en Markov-kedja.*

En Markov Chain Monte Carlo-metod (MCMC) är alltså uppbyggd av Markov-kedjor och Monte Carlo integrering. Viktbaserad sampling är en del av en MCMC-metod. Genom att tillämpa Bayes sats så kan metoden ge mer vikt åt de samplade värden som presterar bättre. Metoden föredrar sådana värden vid uppdatering

av distributionen och kan på så vis begränsa osäkerheten eller variansen hos parametrarna. För en mer utförlig behandling av begreppen, se t.ex. (Gilks *et al.* , 1995).

2.2.2 Algoritm

EPPEs-algoritmen (Laine *et al.* , 2012) bygger på en sorts Markov Chain Monte Carlo-metod med viktbaserad sampling i uppdateringen av parameterdistributionen. Den viktbaserade samplingen grundar sig på att det finns en trovärdighetsfunktion (*eng.* likelihood function) som jämför modelldata med observationer och beaktar detta vid uppdateringen av parameterdistributionen.

EPPEs utgår från ett ensemblesystem med en modell som innehåller ett givet antal parametrar. Felaktiga, eller icke-optimala, parametrar ger upphov till systematiska fel i prognoserna och det finns skäl att optimera dem så att det systematiska felet från parametrarna minskar. Utgångspunkten är att varje parameter θ_l är distribuerad enligt en given distribution, a priori distributionen. Parametrarna kan vara flera till antalet och skrivs i vektorform som $\theta = (\theta_1 \dots \theta_L)$ och distributionen för dem som $\theta \sim N(\mu, \Sigma) = p(\theta)$, där μ är väntevärde och Σ är kovariansmatrisen. Målet för algoritmen är att optimera värdet på μ och Σ genom att ta i beaktande trovärdigheten för parametrarna. Trovärdigheten kan beskrivas som sannolikheten för en enskild prognos med parametervärdet θ att uppnå det observerade värdet y , vilket i korthet kan skrivas som $p(y|\theta)$. A posteriori distributionen, $p(\theta|y)$, är distributionen för parametrarna efter att informationen från observationerna är tillgänglig. Enligt Bayes sats gäller

$$p(\theta|y) \propto p(\theta)p(y|\theta),$$

och utgående från a posteriori är det möjligt att uppdatera distributionen för parametrarna. Uppdateringen av parameterdistributionen följer en given metod.

Algoritmen följer en hierarkisk metod där även osäkerheten för μ och Σ tas i beaktande. Parametrarna är L till antalet så att $\theta \in \mathbb{R}^L$ och de antas vara normalfördelade med väntevärdet μ och kovariansmatris Σ . Fördelningarna för θ , μ och Σ har följande relation

$$\theta_i \sim N(\mu, \Sigma), \tag{2.5a}$$

$$\mu \sim N(\mu_{i-1}, W_{i-1}), \tag{2.5b}$$

$$\Sigma \sim iWish(\Sigma_{i-1}, n_{i-1}), \tag{2.5c}$$

där $\theta_i \in \mathbb{R}^L$ är en parametervektor, $\mu \in \mathbb{R}^L$ är motsvarande väntevärdesvektor och $\Sigma \in \mathbb{R}^{L \times L}$ är parametrarnas kovariansmatris. Tillsammans med n och W utgör W, n, μ och Σ hyperparametrarna i EPPES-algoritmen. Σ är fördelad enligt den inversa Wishart-distributionen (Wishart, 1928) och μ är normalfördelad med väntevärde μ_{i-1} och kovariansmatris W_{i-1} . De här hyperparametrarna uppdateras enligt följande schema

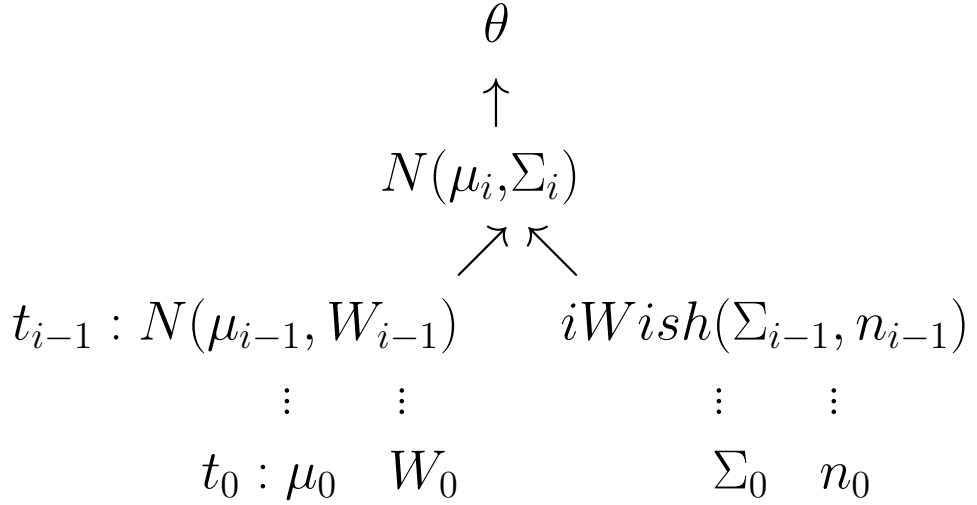
$$W_i = (W_{i-1}^{-1} + \Sigma_{i-1}^{-1})^{-1} \quad (2.6a)$$

$$\mu_i = W_i(W_{i-1}^{-1}\mu_{i-1} + \Sigma_{i-1}^{-1}\theta_{m,i-1}) \quad (2.6b)$$

$$n_i = n_{i-1} + 1 \quad (2.6c)$$

$$\Sigma_i = \frac{(n_{i-1}\Sigma_{i-1} + (\theta_{m,i-1} - \mu_i)(\theta_{m,i-1} - \mu_i)^T)}{n_i}, \quad (2.6d)$$

där $\theta_{m,i-1} = (\theta_{m,1} \dots \theta_{m,L})_{i-1}$ är en parametervektor för ensemblemedlem m vid iteration $i - 1$. De nya hyperparametrarna W, μ, n och Σ är medelvärde av W_i, μ_i, n_i respektive Σ_i .



Figur 2.3: Illustration av EPPES parameteruppdateringsschema. EPPES följer en hierarkisk metod som utgår från hyperparametrarna μ_0, W_0, Σ_0 och n_0 . De samplade parametervärdena θ är distribuerade enligt $N(\mu_i, \Sigma_i)$, där μ_i och Σ_i beror på tidigare värden på hyperparametrarna. Parametervektorerna θ samplas från den senast uppdaterade distributionen $N(\mu_i, \Sigma_i)$.

Uppdateringsschemat för EPPES hyperparametrar illustreras i figur 2.3. Vid initialisering får algoritmen värdet på fyra hyperparametrar: μ_0, Σ_0, W_0 och n_0 . Hyperparametrarna uppdateras efter att ny information om parametervärdena θ

finns tillgänglig.

I sin helhet beskrivs algoritmen som

1. Initialisera hyperparametrarna μ_0, Σ_0, n_0 och W_0 . Bestäm antal ensemblemedlemmar M
2. Sampla M parametrar θ från a priori distributionen $p(\theta) = N(\mu, \Sigma)$
3. Kör en ensembleprognos med de dragna parametrarna
4. Beräkna trovärdigheten $p(y|\theta)$ för varje ensemblemedlem
5. Omsampla (*eng.* re-sample) parametervärdena
6. Uppdatera parameterdistributionen enligt uppdateringsschemat (2.6) för att få a posteriori distributionen $p(\theta|y)$. Gå tillbaka till punkt 2 och fortsätt tills konvergens uppnås. Den nya a priori distributionen är den gamla a posteriori distributionen.

EPPEs-algoritmen utgår från hyperparametrarna μ, Σ, W och n , vilka väljs utgående från den information som finns att tillgå om parametrarna och deras troliga fördelning. Från den givna distributionen $N(\mu, \Sigma)$ dras lika många parametervektorer θ som det finns ensemblemedlemmar. Idén är att varje ensemblemedlem har förutom avvikande initialvärden även avvikande parametervärden. Vid nästa steg körs ensembleprognosen varefter trovärdighetsfunktionen beräknas mellan det observerade värdet och varje ensemblemedlem. Baserat på dess värden omsamlas parametrarna så att parametervärden med högre sannolikhet får större betydelse vid uppdatering av hyperparametrarna. Till sist uppdateras hyperparametrarna och algoritmen går tillbaka till punkt 2. Då algoritmen går tillbaka till punkt 2 så kommer en ny ensemble med nya initialvärden att användas. På det här sättet är det möjligt att hitta sådana parametervärden vilka fungerar genomsnittligt bra och algoritmen tar på så sätt i beaktande att eventuella parametrar beror på väderslag, årstid och tidpunkt.

Trovärdighetsfunktionen väljs så att ett lämpligt antal parametrar accepteras vid omsamlingen. Trovärdighetsfunktionen är proportionell mot exponentialfunktionen av ett kvadratisk uttryck

$$\begin{aligned} p(y|\theta) &\propto \exp\left(-\frac{1}{2}(y - F(x, \theta))^T \Sigma_{obs}^{-1}(y - F(x, \theta))\right) \\ &\propto \exp(F_{kostnad}(x)), \end{aligned}$$

där y är det observerade värde, $F(x, \theta)$ det prognostiserade värdet med initialvärdet x och parametervektor θ och Σ_{obs} är kovariansmatrisen för observationsfelet. Det här gör att kostnadsfunktionen bör väljas på lämpligt sätt för att få rimliga värden på trovärdighetsfunktionen. Laine *et al.* (2012) nämner ingen speciell funktion för trovärdighetsfunktionen, men tillägger att det är viktigt att trovärdighetsfunktionen kan räknas unikt för att kunna skilja bra parametervärden från dåliga värden. En fullständig beskrivning av algoritmen finns i (Laine *et al.* , 2012).

Kapitel 3

Numeriska experiment

Det här kapitlet behandlar de numeriska experimenten samt de numeriska modeller som används i experimenten. Först beskrivs de numeriska modellerna av Lorenz96 och EPPES, därefter följer en beskrivning av experimentupplägget för de båda experimenten och till sist följer ett exempel med beräkningar.

3.1 Numeriska modeller

De numeriska modellerna för Lorenz96 och algoritmen EPPES är förhållandevis enkla. Lorenz96-modellen utgår från ett initialvärde X_{t_0} från vilket följande värde X_{t_1} fås genom att numeriskt integrera någondera av ekvationerna för L96, se ekvationerna (2.1), (2.2) och (2.3). Den numeriska integreringen utförs med Runge-Kutta4-metoden (Lambert, 1973, s. 120) för att få en tillräcklig noggrannhet i integrationen. Runge-Kutta4-metoden löser differentialekvationen $dx/dt = f(x, t)$ med initialvärde $x(t_0) = x_0$ enligt följande princip:

$$\begin{aligned}x_{n+1} &= x_n + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\t_{n+1} &= t_n + \Delta t \\k_1 &= f(t_n, x_n) \\k_2 &= f(t_n + \frac{\Delta t}{2}, x_n + \frac{\Delta t}{2}k_1) \\k_3 &= f(t_n + \frac{\Delta t}{2}, x_n + \frac{\Delta t}{2}k_2) \\k_4 &= f(t_n + \Delta t, x_n + \Delta tk_3),\end{aligned}$$

där Δt motsvarar tidssteget mellan x_n och x_{n+1} . För att få mer exakta integreringar är tidssteget litet. I fallet med Lorenz96 så ersätts $f(x, t)$ med någon

av ekvationerna som beskriver dynamiken och x_n motsvarar tiden vid t_n . Den fullständiga L96-modellen används för att generera syntetiska observationer och den parametriserade versionen för prognoser. I båda fallen används ett tidssteg på 0,0025 tidsenheter.

EPPEs-algoritmen följer exakt den beskrivna algoritmen. Det som krävs för att kunna utföra de numeriska experiment är någon form av sampling av slumpmässiga tal samt någon metod för att multiplicera och invertera matriser. Den numeriska algoritmen finns i sin helhet på: <http://helios.fmi.fi/~lainema/eppes/>.

EPPEs

1. Initialisera μ, Σ, n, W
2. Sampla lika många parametervektorer som antal ensemblemedlemmar $m = 1, \dots, M$

$$\theta_m = (\theta_{m,1} \quad \dots \quad \theta_{m,L})$$

3. Ensembleprognos med Lorenz96:

Lorenz96

- (i) Initialvärden från observationer genererade med den fullständiga modellen
- (ii) Ensembleprognos med den parametriserade versionen av L96 (parametervärden från samplingen), M ensemblemedlemmar
- (iii) Output: en för varje ensemblemedlem $m = 1, \dots, M$

4. Beräkna kostnadsfunktionen för varje medlem: jämför det observerade värdet med modellvärdet för att få fram a posteriori distributionen
5. Omsampla parametrarna, d.v.s. ge mer vikt åt bra parametervärden
6. Uppdatera μ, Σ, n, W . Gå tillbaka till punkt 2.

Figur 3.1: Schematisk bild över EPPEs-algoritmen med Lorenz96, vilken ger en överblick över hur algoritmen fungerar med Lorenz96-modellen. Lorenz96-modellen är inbäddad som en fristående enhet i EPPEs-algoritmen.

Figur 3.1 ger en överblick av algoritmen då ensembleprognoserna utförs med L96-modellen. I figuren framgår det att Lorenz96-modellen är en oberoende enhet, vilken enbart behöver de samplade parametervärdena från EPPES. De samplade parametervärdena används vid ensembleprognosen vars initialvärden baseras på syntetiska observationer. De syntetiska observationerna genereras med den fullständiga Lorenz96-modellen och används förutom vid initialiseringen även vid evaluering av kostnadsfunktionen. Kostnadsfunktionen beräknas för varje ensemblemedlem för att få ett konkret svar på hur väl ett visst parametervärde presterar. Funktionen väljs så att varje parametervärde betraktas som unikt och så att omsamplingen accepterar ett lämpligt antal parametrar.

3.2 Experimentupplägg

Experimentupplägget för de numeriska experimenten följer samma upplägg som Laine *et al.* (2012) beskriver. De använder sig av en ensemble på 50 medlemmar, medan ensemblestorleken i den här studien varierar från två till 50. I övrigt är upplägget detsamma. De experiment som utförs med en ensemble på 50 medlemmar används för att kontrollera och jämföra tidigare resultat (Laine *et al.* , 2012).

De syntetiska observationerna genereras med den fullständiga modellen med ett tidssteg på 0,0025 tidsenheter i den numeriska integreringen. Den fullständiga modellen har 40 stycken långsamma variabler, 8 stycken snabba variabler per långsam variabel, forceringstermen för de snabba variablerna F_u är 10, forceringstermen för de långsamma variablerna F är 10 och de övriga parametrarna b , c och h är 10, 10 respektive 1.

Ensembleprognoserna beräknas med den parametriserade versionen av L96-modellen som beskrivs i ekvation (2.3). Den parametriserade versionen har 40 långsamma variabler och forceringstermen F är 10. Initialvärden för ensemblen är litet avvikande värden från det analyserade värdet x_{analys} som i sin tur fås genom att lägga till en liten avvikelse till det observerade värdet x_{obs} . Här används slumpmässiga, normalfördelade tal för att få det analyserade värdet x_{analys} och initialvärdena x_{init} :

$$\begin{aligned} x_{\text{analys}} &= x_{\text{obs}} + \epsilon, \quad \epsilon \sim N(0, \sigma_{\text{analys}}^2), \text{ och} \\ x_{\text{init}} &= x_{\text{analys}} + \epsilon, \quad \epsilon \sim N(0, \sigma_{\text{pert}}^2). \end{aligned}$$

Det analyserade värdet x_{analys} är detsamma för samtliga ensemblemedlemmar och

det enda som skiljer ensemblemedlemmarna från varandra är det slumpmässiga tal som läggs till x_{analys} . Här är $\sigma_{pert} = 0,1$ och $\sigma_{analys} = 0,05$.

Kostnadsfunktionen beräknas mellan observationen x_{obs} och en ensemblemedlem $x_{modell,m}$ för att få fram ett värde på prognosförmågan. Ju lägre värden kostnadsfunktionen antar, desto närmare är det prognostiserade värdet det observerade värdet och desto bättre är parametervärdena. Varje ensemblemedlem har ett initialvärde x_{init} och en parametervektor $\theta = (\theta_1 \theta_2)$. Med hjälp av prognosmodellen (ekvation (2.3)) fås ett resultat $x_{modell,m}$ för varje ensemblemedlem m . Kostnadsfunktionen beräknas mellan det observerade värdet och det värde som modellen ger enligt följande princip

$$\begin{aligned} f(x_{modell,m}) &= (x_{modell,m} - x_{obs}) \Sigma_{obs}^{-1} (x_{modell,m} - x_{obs})^T \\ &= \begin{pmatrix} x_{m,1} - x_{o,1} & \dots & x_{m,K} - x_{o,K} \end{pmatrix} \begin{pmatrix} \sigma_1^{-2} & & 0 \\ & \ddots & \\ 0 & & \sigma_K^{-2} \end{pmatrix} \begin{pmatrix} x_{m,1} - x_{o,1} \\ \dots \\ x_{m,K} - x_{o,K} \end{pmatrix} \\ &= \sum_{k=1}^K \sigma_k^{-2} (x_{m,k} - x_{o,k})^2. \end{aligned}$$

Då felet för analysen är detsamma för varje variabel så kan ovanstående förenklas till

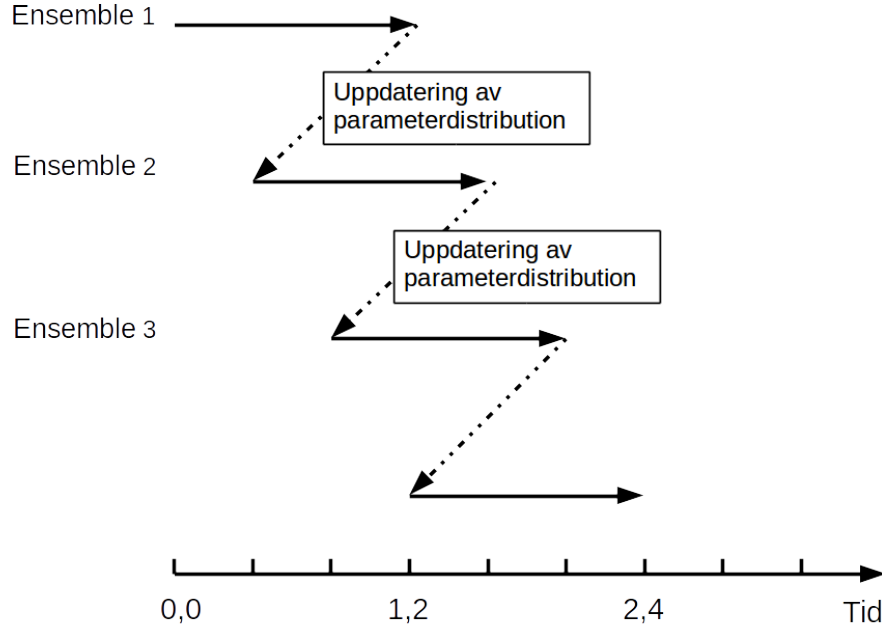
$$f(x_{modell,m}) = \sigma_{analys}^{-2} \sum_{k=1}^K (x_{m,k} - x_{o,k})^2, \quad (3.1)$$

där $x_{m,k}$ är den m :te ensemblemedlemmens värde i variabel k , $x_{o,k}$ är det observerade värdet för variabel k och σ_{analys} är observationsfelet för x . Därtill behövs ytterligare en finjustering av kostnadsfunktionen för att få en bra acceptansprocent i EPPES-algoritmen, vilket kan göras genom att multiplicera värdet på $f(x_{modell})$ med en lämplig koefficient. I det här fallet ger funktionen väldigt höga värden vilket inte är lämpligt vid omsamplingen och det är därför nödvändigt att skala ner resultatet för att få mer rimliga resultat vid omsamplingen. I det här fallet multipliceras kostnadsfunktionens resultat med $1/4000$ så att själva kostnadsfunktionen ser ut som följer

$$f(x_{modell,m}) = \frac{\sigma_{analys}^{-2}}{4000} \sum_{k=1}^K (x_{m,k} - x_{o,k})^2 = 0,1 \sum_{k=1}^K (x_{m,k} - x_{o,k})^2.$$

En dåligt vald kostnadsfunktion kan göra att EPPES även accepterar sämre

parametervärden, vilket i sin tur leder till att konvergensen blir långsam eller till och med uteblir. Det är även värt att komma ihåg att algoritmen strävar till att minimera värdet på kostnadsfunktionen, vilket innebär att parametervärdena optimeras enligt den valda kostnadsfunktionen. Här används en form av kvadrerat medelvärde över tillståndsvariablerna X_k , d.v.s. det kvadrerade avståndet mellan observationen och prognosen minimeras. Kostnadsfunktionen påverkar således hur parametrarna optimeras och det är därför skäl att undersöka vilken sorts kostnadsfunktion som passar för den numeriska modell som används.



Figur 3.2: *Experimentupplägget grundar sig på att varje ensembleprognos beräknas 1,2 tidsenheter framåt, varefter parameterdistributionen uppdateras. Följande ensembleprognos startar 0,4 tidsenheter efter den föregående och använder den uppdaterade parameterdistributionen vid samplingen av parametrarna. Den här formen av upplägg fungerar enbart som en form av efteranalys och inte i realtid. I det här fallet genereras observationerna på förhand, vilket möjliggör den här formen av experimentupplägg.*

Varje ensembleprognos beräknas 1,2 tidsenheter framåt varefter kostnadsfunktionen beräknas mellan observationen och varje ensemblemedlem. Poängen från kostnadsfunktionen tas i beaktande då EPPES omsamlar och uppdaterar μ , n , W och Σ . Därefter är det möjligt att sampla nya parametervärden och köra en ny ensembleprognos. Den nya ensembleprognosen startar 0,4 tidsenheter efter den föregående. Det hela illustreras i figur 3.2.

EPPES-algoritmens initialvärden är för de båda experimenten $W = \begin{pmatrix} 100 & 0 \\ 0 & 100 \end{pmatrix}$, $n = 10$ vilken hålls konstant och $\mu = (0,5, 0,5)$. I det första experimentet har kova-

riansmatrisen högre värden $\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, och i det andra experimentet är kovariansmatrisen lägre, $\Sigma = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}$. Då kovariansmatrisen har högre värden betyder det att det finns en större osäkerhet kring de förmodade parametervärdena medan lägre värden betyder att det finns anledning att tro att de förmodade parametervärden är nära de optimala värdena. Matrisen W beskriver hur stor osäkerhet som finns kring μ ; ju högre värden W har, desto mer osäkerhet finns det i värdet på μ . Värdet på n beskriver osäkerheten kring kovariansmatrisen Σ . I det här fallet är n konstant och osäkerheten för Σ hålls därmed på samma nivå genom hela algoritmen. Ifall inga restriktioner finns på n och W , så kommer n att gå mot oändligheten och W mot nollmatrisen då iterationsantalet växer. En sammanfattning av initialvärdena för Lorenz96 och EPPES finns i tabell 3.1 respektive tabell 3.2.

	F	F_u	K	J	b	c	h
Fullständig modell	10	10	40	8	10	10	1
Prognosmodell	10	-	40	-	-	-	-

Tabell 3.1: Initialvärden för Lorenz-96.

	μ_0	Σ_0	n	W	n_{max}
Experiment 1	$\begin{pmatrix} 0.5 & 0.5 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	10	$\begin{pmatrix} 100 & 0 \\ 0 & 100 \end{pmatrix}$	10
Experiment 2	$\begin{pmatrix} 0.5 & 0.5 \end{pmatrix}$	$\begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}$	10	$\begin{pmatrix} 100 & 0 \\ 0 & 100 \end{pmatrix}$	10

Tabell 3.2: Initialvärden för EPPES för de olika experimenten.

3.2.1 Exempel - uträkningar

Det här exemplets uppgift är att förtydliga hur algoritmens uträkningar fungerar samt ge exempel på vad en liten ensemble innebär. Exemplet utgår från initialvärdena för experiment 1, se tabell 3.2. Initialvärdena för L96 finns i tabell 3.1. I det här exemplet är ensemblestorleken tre.

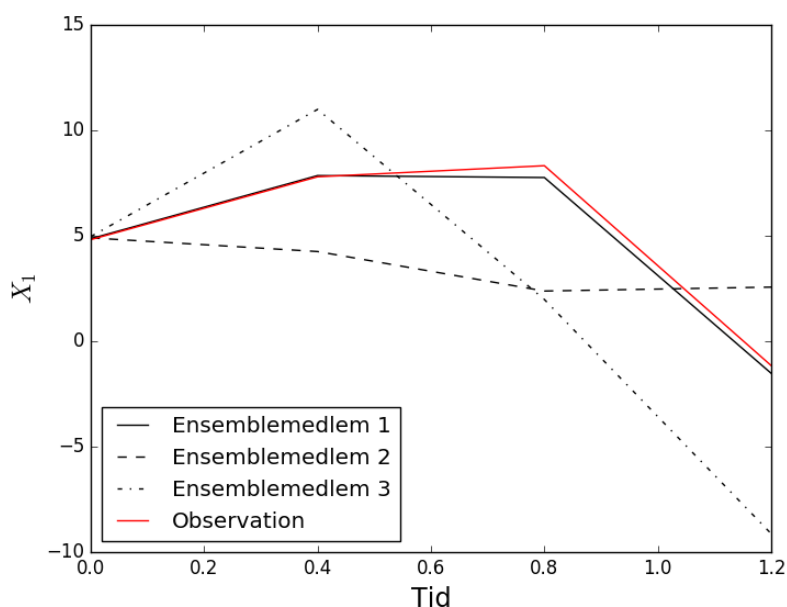
Det första steget är att sampla lika många parametervektorer som det finns ensemblemedlemmar. I tabell 3.3 finns värdena på de samplade parametrarna för varje ensemblemedlem.

Följande steg är att göra en ensembleprognos, där varje ensemblemedlem har de parametervärden som samplades i steg 1. Ensemblemedlemmarna har precis som i en normal ensembleprognos olika initialvärden vilka är tagna runt det

Ensemblemedlem	Parameter 1	Parameter 2	$F_{kostnad}(X)$	Omsampling
1	2,4096	0,02056	15,2	Ja
2	-0,5483	2,6385	48,4	Nej
3	-0,6899	-0,5857	285,0	Nej

Tabell 3.3: Samplade parametervärden samt värdet på kostnadsfunktionen. Den sista kolumnen visar om parametervärdet omsamlas, d.v.s. ifall parametervärdet tas i beaktande vid uppdatering av hyperparametrarna.

analyserade värdet. Figur 3.3 visar värdet på (t, X_1) för hela ensemblen (svarta linjer) samt observationen (röd linje). Baserat på den här figuren så kan vi anta att ensemblemedlem 1 kommer att vara den medlem vars kostnadsfunktion har lägst värden.



Figur 3.3: Exempel på hur en ensembleprognos ser ut med olika värden på parametrarna. Notera att ensemblemedlemmarna har olika initialvärden. Den röda linjen representerar observationen och de svarta linjerna ensemblemedlemmarnas prognoser, där linjens stil visar vilken ensemblemedlem som är vilken. Prognosmodellens och observationens data-serier finns med 0,4 tidsstegs intervall och jämförelsen mellan prognos och observation sker vid 1,2 tidsenheter. Vid beräkningarna används ett mindre tidssteg, $\Delta t = 0,0025$ tidsenheter, men för att spara lagringsutrymme sparas en mindre mängd data. Av den anledningen förefaller ovanstående modelldata mer diskret än vad den är.

I steg 3 jämförs observationen med prognosen för varje ensemblemedlem. I det här fallet beräknar vi en kostnadsfunktion enligt ekvation (3.1) för att få fram hur bra en prognos motsvarar det observerade värdet. På så vis får vi även en uppfattning hur bra parametervärdena var. Kostnadsfunktionens värden för

samtliga ensemblemedlemmar finns i fjärde kolumnen i tabell 3.3.

Följande steg, steg 4, är att omsampla parametrarna, d.v.s. baserat på kostnadsfunktionens värden så kommer en eller flera ensemblemedlemmars parametrar att accepteras som bra. Dessa omsamplade värden tas i beaktande vid uppdatering av hyperparametrarna. Här omsamplades enbart ensemblemedlem 1, vilket vi också antog baserat på figur 3.3.

Det sista steget är att uppdatera hyperparametrarna W, n, μ och Σ . Vid uppdateringen av hyperparametrarna tas enbart i beaktande de omsamplade värdena, vilket i det här fallet innebär att enbart parametervektorn från ensemblemedlem 1 tas med. I de fall där flera parametervärden omsamlas beräknas μ, Σ, W och n för varje ensemblemedlem varefter de nya hyperparametrarna är medelvärde över respektive hyperparameter. Uträkningarna grundar sig på uppdateringsschemat från ekvation (2.6).

Värdet på W_1 beräknas först:

$$\begin{aligned} W_1 &= (W_0^{-1} + \Sigma_0^{-1})^{-1} \\ &= \left(\left(\begin{pmatrix} 100 & 0 \\ 0 & 100 \end{pmatrix}^{-1} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}^{-1} \right)^{-1} = \begin{pmatrix} 0,99 & 0 \\ 0 & 0,99 \end{pmatrix}, \end{aligned}$$

därefter beräknas värdet på μ_1

$$\begin{aligned} \mu_1 &= W_1(W_0^{-1}\mu_0 + \Sigma_0^{-1}\theta_1) \\ &= \begin{pmatrix} 0,99 & 0 \\ 0 & 0,99 \end{pmatrix} \left(\begin{pmatrix} 100 & 0 \\ 0 & 100 \end{pmatrix}^{-1} \begin{pmatrix} 0,5 & 0,5 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 2,4096 & 0,02056 \end{pmatrix} \right) \\ &= \begin{pmatrix} 2,39 & 0,025 \end{pmatrix}, \end{aligned}$$

följt av värdet på n : $n_1 = n_0 + 1 = 10 + 1 = 11$, och till sist Σ_1 enligt

$$\begin{aligned} \Sigma_1 &= (n_0\Sigma_0 + (\theta_1 - \mu_1)(\theta_1 - \mu_1)^T)/n_1 \\ &= \frac{1}{11} \left(10 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 2,4096 - 0,5 & \\ 0,02056 - 0,5 \end{pmatrix} \begin{pmatrix} 2,4096 - 0,5 & 0,02056 - 0,5 \end{pmatrix} \right) \\ &= \begin{pmatrix} 0,91 & -0,8E-5 \\ -0,8E-5 & 0,91 \end{pmatrix}. \end{aligned}$$

Kapitel 4

Resultat

Det här kapitlet innehåller resultaten från de experiment som beskrivs i kapitel 3. Den första delen presenterar resultaten från de båda experimenten och den andra delen diskuterar resultaten i relation till hypotesen samt jämför de två experimentens resultat.

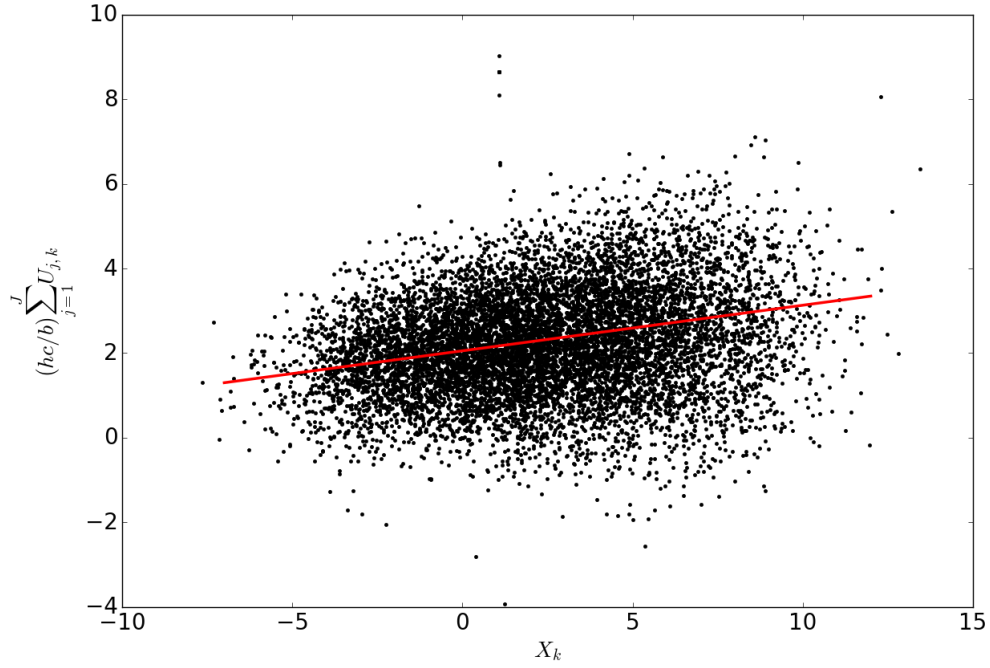
4.1 Experiment med liten ensemble

Resultaten från experimenten med liten ensemble är uppdelade i två delar; experiment 1 med större osäkerhet i initialvärdena för parametrarna och experiment 2 med mindre osäkerhet. Som referensvärden för konvergensen används motsvarande experiment med en ensemble på 50 medlemmar. Dessa värden är möjliga att jämföra med tidigare resultat - dock med resultat från en stokastisk version av Lorenz96 (Laine *et al.* , 2012). Utgångspunkten är att den stokastiska forceringen i Lorenz96 modellen har en liten inverkan på resultat, men det är skäl att påminnas att resultaten inte kommer att stämma överens exakt.

4.1.1 Konvergenstest

Konvergenstestet med en ensemble på 50 medlemmar visar att de optimala parametervärden är ungefär 2,0 för parameter 1 och ungefär 0,1 för parameter 2. Dessa värden används som referensvärden för att kontrollera konvergensen för experimenten med mindre ensemblestorlek. Laine *et al.* (2012) har resultat på 1,8 respektive 0,11, så den stokastiska forceringen har en liten inverkan på resultatet för parameter 1 och till synes ingen inverkan på parameter 2.

Parametriseringen av den forcering som orsakas av de snabba variablerna ger ett genomsnittligt gott resultat. Figur 4.1 visar forceringen av de snabba variab-

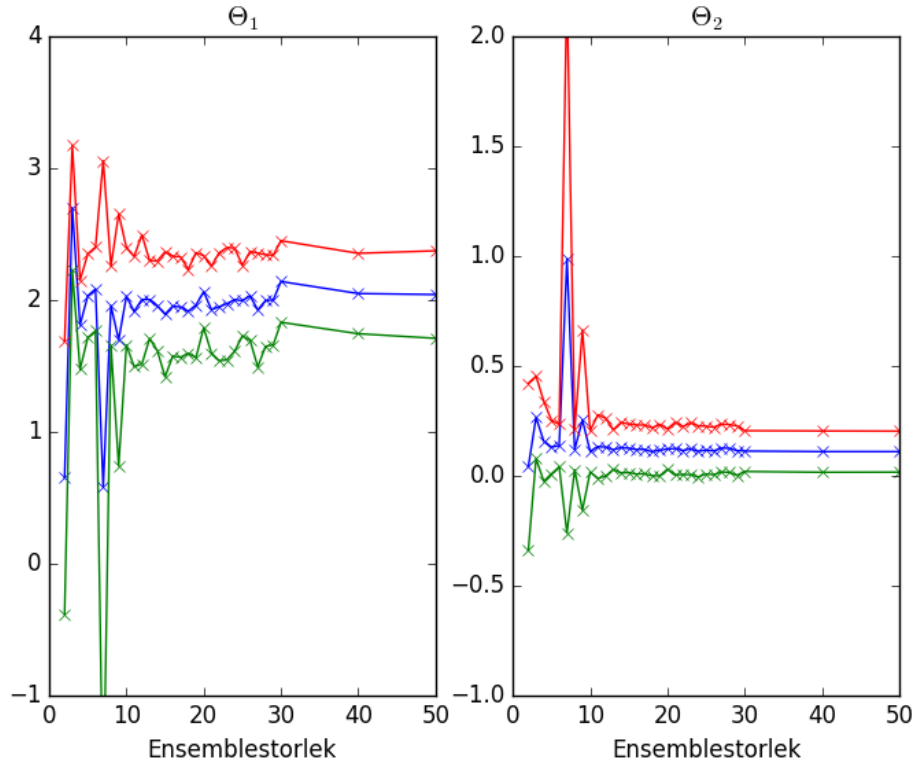


Figur 4.1: De snabba variablerna orsakar en forcering $F_{snabb} = (hc/b) \sum_{j=1}^J U_{j,k}$, vilken parametreras som en linjär regression. De svarta prickarna motsvarar punkterna (X_k, F_{snabb}) och den röda heldragna linjen $(X_k, g(x_k, \theta))$ visar parametreringen av F_{snabb} för de optimala parametervärdena.

lerna som svarta prickar och parametreringen av forceringen som en röd linje. EPPES konvergerade till $\theta = (2,054 \ 0,108)$ i experiment 1 med en ensemble på 50. Den röda linjen visar alltså den linje som motsvarar parametreringsschemat i ekvation (2.4): $2,054 + 0,108X_k$. Diagrammet visar att parametreringen fungerar samt att algoritmen söker efter parametervärden oberoende av tillståndsvariablernas värden. Diagrammet bekräftar att kostnadsfunktionen är väl definierad och att koefficienten i funktionen är i rätt storleksklass, eftersom resultatet här motsvarar tidigare resultat, se (Laine *et al.*, 2012). Ifall kostnadsfunktionen inte är väl definierad så kommer inte heller optimeringen av parametrarna att fungera.

4.1.2 Experiment 1 - mer optimal a priori distribution

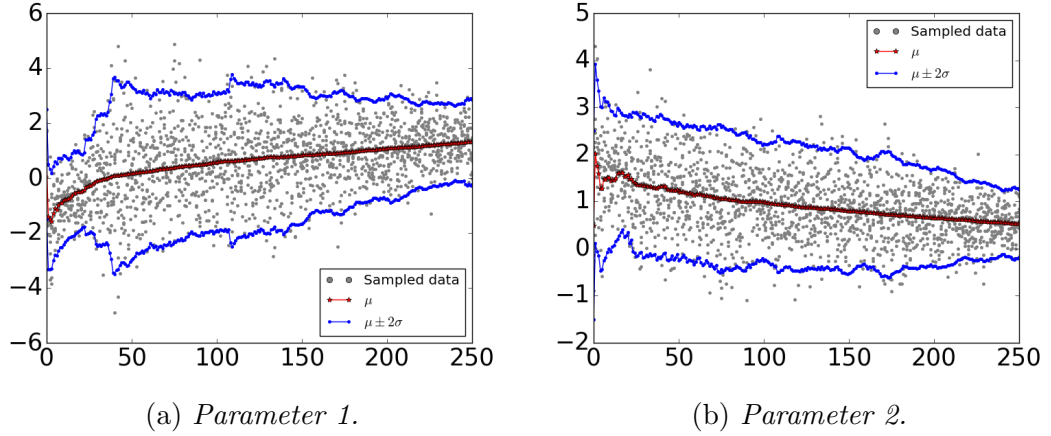
Det första experimentet med små ensembler har en större osäkerhet i a priori distributionen för parametrarna. A priori distributionen har väntevärdet $\mu_0 = (0,5 \ 0,5)$ och kovariansmatrisen har värdet $\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. Ensemblestorleken varierar mellan två och 50.



Figur 4.2: *Experiment 1. Distributionen av parametervärden som en funktion av ensemblestorlek efter 100 steg med algoritmen. Det vänstra diagrammet visar resultatet för parameter 1 och den högra för parameter 2. I båda diagrammen representerar den blå linjen medelvärdet μ , den gröna linjen övre gränsen för 95% ($\mu + 2\sigma$) och den röda linjen nedre gränsen för 95% ($\mu - 2\sigma$).*

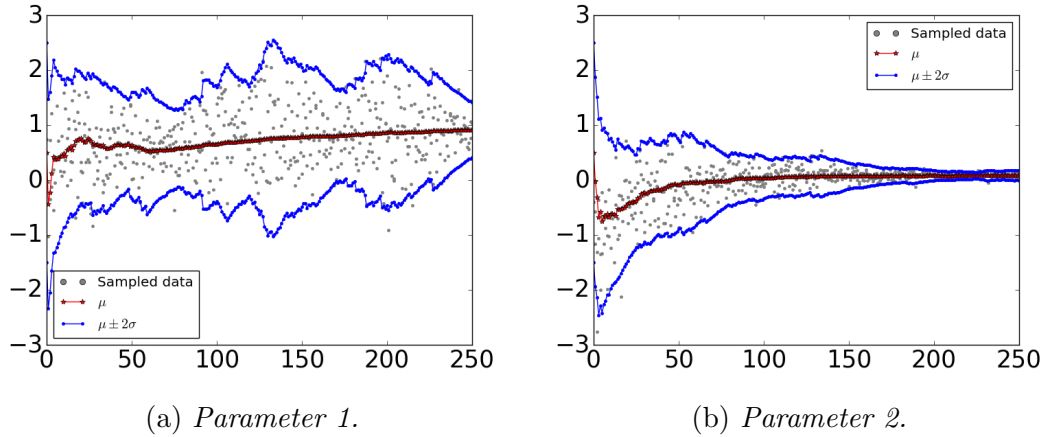
Figur 4.2 presenterar resultatet från experiment 1 som en funktion av ensemblestorlek efter 100 iterationer. Det vänstra diagrammet visar resultatet för parameter 1 som funktion av ensemblestorlek och det högra diagrammet visar värdet för parameter 2 som funktion av ensemblestorlek. Den blåa linjen representerar medelvärdet för respektive parameter. Den röda och gröna linjen visar var gränsen för 95%-sannolikhet (2 standardavvikelser från medelvärdet) går. Den här figuren visar att efter 100 steg med EPPES är det svårt att särskilja resultaten från de små ensemblerna från de stora. Det finns dock två tydliga avvikelser i figuren; resultatet från ensemblen med två respektive sju medlemmar. De övriga ensemblestorlekarna är nära de optimala värden för respektive parameter, vilket tyder på att ensemblestorleken i det här fallet har en liten betydelse för konvergensen.

Figur 4.3 visar resultatet från experiment 1 med en ensemble på sju medlemmar. Diagrammen visar att algoritmen antar fel värden i början, vilket sedan korrigeras för att sedan närma sig det optimala värdet runt 2,0 respektive 0,1. Efter 250 iterationer har parametervärdena närmast sig 1,315 respektive 0,539,



Figur 4.3: *Parameteroptimering med en ensemble på sju medlemmar. Figur (a) och (b) visar utvecklingen av parametervärden som funktion av iterationer för parameter 1 respektive parameter 2.*

men kovariansmatrisen har ännu stora värden vilket gör att de optimala värdena ändå återfinns inom den distribution som EPPES har nått efter 250 iterationer.

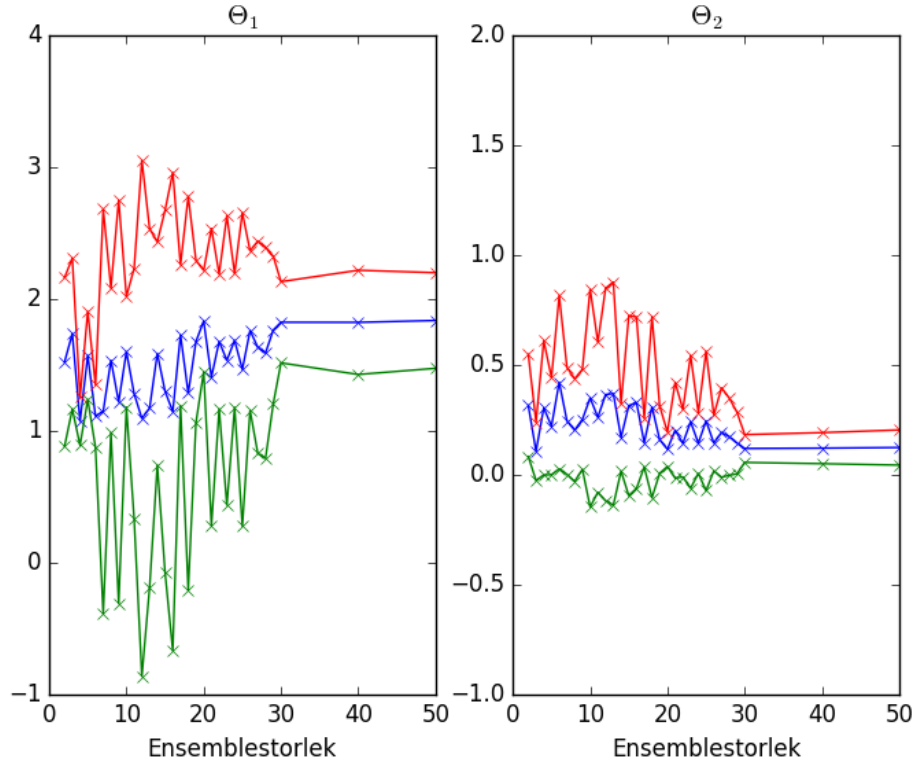


Figur 4.4: *Experiment 1. Parameteroptimering med en ensemble på två medlemmar. Figur (a) och (b) visar utvecklingen av parameterdistributionen som funktion av iterationer för parameter 1 och parameter 2.*

För en ensemble med två medlemmar ter sig resultatet annorlunda. Figur 4.4 visar att algoritmen har svårigheter att få parameter 1 att konvergera, även om algoritmen går mot rätt håll. Parameter 2 har inte samma problem utan kommer mycket nära det optimala värdet. Efter 250 iterationer har värdet på parametrarna nått 0,914 respektive 0,085. En ensemble på enbart två medlemmar är i det här fallet för litet för att EPPES skall kunna konvergera.

4.1.3 Experiment 2 - mindre optimal a priori distribution

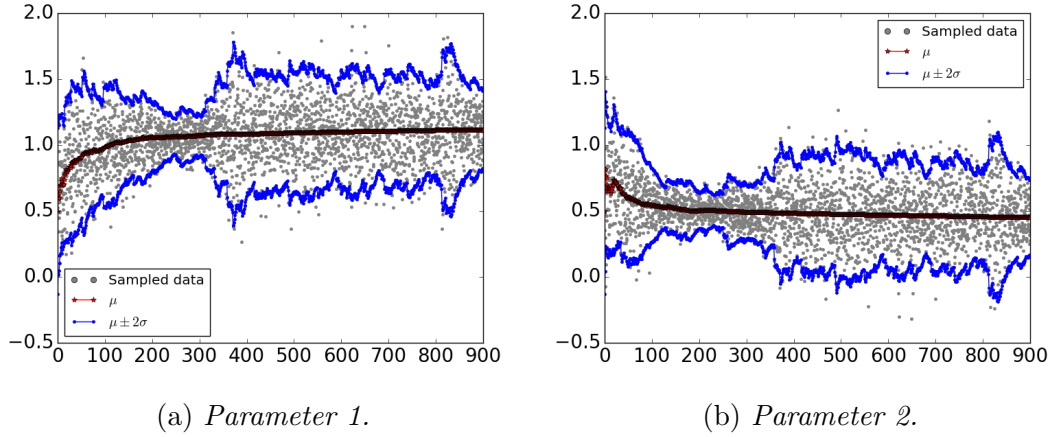
Det andra experimentet med liten ensemblestorlek har en mindre osäkerhet i a priori distributionen än det första experimentet har. I det här experimentet har a priori distributionen väntevärde $\mu = (0,5, 0,5)$ och en kovariansmatris $\Sigma = \begin{pmatrix} 0,1 & 0 \\ 0 & 0,1 \end{pmatrix}$. I det här fallet innebär det att det optimala värdet på parametrarna inte finns inom ramarna för a priori distributionen.



Figur 4.5: *Experiment 2. Distributionen av parametervärden som en funktion av ensemblestorlek efter 100 iterationer. Det vänstra diagrammet presenteras värdena för parameter 1 som funktion av ensemblestorlek och i det högra värdena för parameter 2. I båda diagrammen motsvarar den blå linjen μ , den gröna $\mu + 2\sigma$ och den röda $\mu - 2\sigma$.*

Figur 4.5 visar resultatet för parametervärdena som funktion av ensemblestorlek efter 100 iterationer. Diagrammen visar väntevärdet för respektive parameter samt 95%-marginalerna för ensemblestorlekar upp till 50 för parameter 1 (vänster bild) och för parameter 2 (höger bild). Diagrammen visar att båda parametrarna konvergerar långsammare mot det optimala värdet än de gjorde i det första experimenten (jämför Figur 4.2). Den smala initialdistributionen gör det svårt för EPPES att sampla parametervärden utanför den givna distribution, vilket påverkar konvergensen för båda parametrarna. En större ensemblestorlek ökar sannolikheten för att få värden utanför den givna distributionen och därför kon-

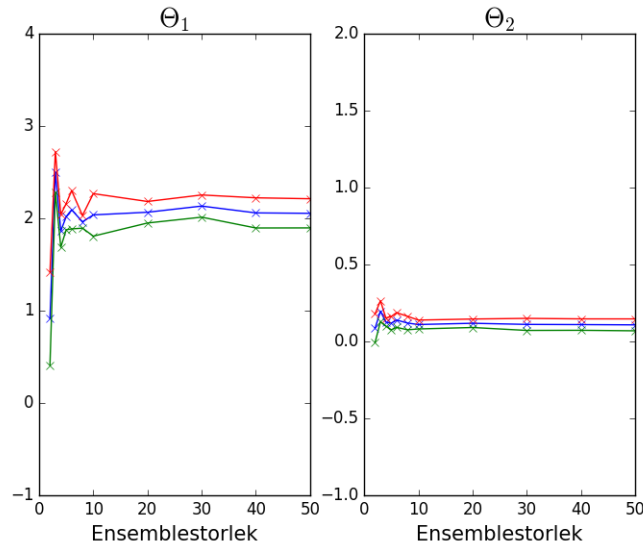
vergerar EPPES bättre då ensemblestorleken är stor. En mindre ensemblestorlek kräver fler iterationer för att konvergera, men garanterar inte konvergens av parametrarna. I det här fallet är initialdistributionen för snäv för att EPPES skall kunna konvergera med en liten ensemble, vilket figur 4.6 visar. Diagrammen visar resultatet för en ensemble på tre medlemmar, där algoritmen körs 900 iterationer framåt utan att nå konvergens. Det är tydligt att en ensemble på tre medlemmar inte räcker till i det här fallet.



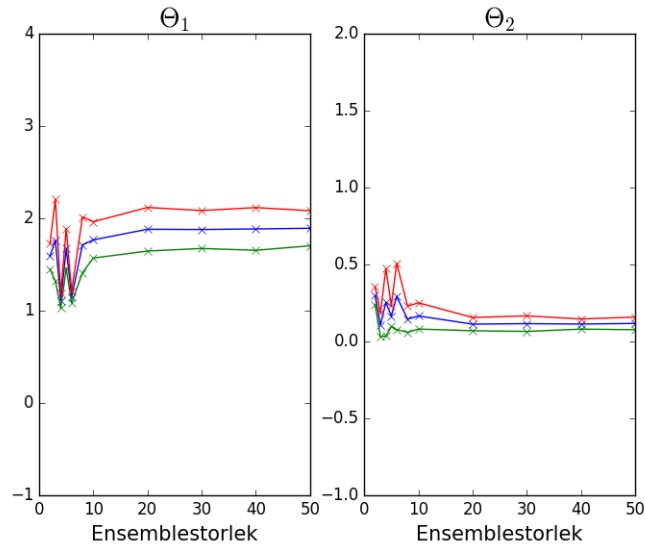
Figur 4.6: Exempel på hur utvecklingen av parametervärdena ser ut för en ensemble på tre medlemmar för experiment 2. Parameter 1 uppdateras åt rätt håll, men inte ens efter 900 iterationer har värdet för parameter 1 konvergerat. Parameter 2 har en liknande utveckling som parameter 1, men det optimala värdet finns innanför 95%-gränserna.

4.2 Verifiering av hypotesen

Det första experimentet har en initialdistribution där parametrarna har större osäkerhet och det andra experimentet har en initialdistribution där osäkerheten är betydligt mindre. I det andra experimentet är distributionens standardavvikelse en tiondel av vad den var i det första experimentet. Figur 4.2 visar parametervärdena som funktion av ensemblestorlek för experiment 1 och figur 4.5 motsvarande resultat för experiment 2. Det finns några tydliga skillnader i diagrammen: (1) väntevärdet är detsamma för nästan alla ensemblestorlekar i experiment 1 och för experiment 2 har väntevärdet en krokigare linje, (2) distributionen är smalare i experiment 1 än i experiment 2 detta trots att experiment 2 hade en snävare initialdistribution och (3) i experiment 2 verkar det som att algoritmen försöker kompensera med att öka osäkerheten och på så vis nå en distribution där det optimala värdet återfinns.



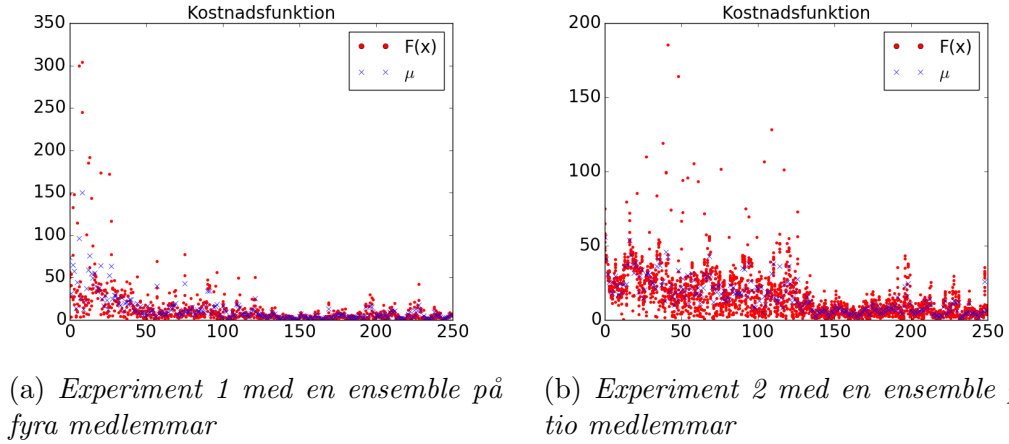
Figur 4.7: *Experiment 1. Efter 250 iterationer har alla ensembler förutom de allra minsta med två och tre medlemmar konvergerat. En ensemble på tre medlemmar avviker i värdet på parameter 1, vilket gör att minsta möjliga ensemble är fyra. Det kan hända att avvikelsen är slumpmässig och att det således skulle vara möjligt med en ensemble på tre medlemmar. Det kräver dock fler experiment för att bekräfta.*



Figur 4.8: *Experiment 2. Efter 250 steg med algoritmen så har de ensembler som har tio eller fler medlemmar konvergerat till ett värde nära det optimala. För de mindre ensemblerna är det svårt att säga vilka som har konvergerat och det är säkrast att dra gränsen för minsta möjliga ensemblestorlek vid tio medlemmar.*

Genom att ta en titt på motsvarande resultat efter 250 iterationer får vi en bredare bild över resultaten. Figur 4.7 visar resultatet för experiment 1 och figur 4.8 för experiment 2. För experiment 1 så har alla ensembler med storlek fyra och större konvergerat till de optimala värdena, medan alla ensembler på tio eller större har konvergerat för experiment 2. Diagrammen visar att skillnaden är

mindre mellan resultaten efter 250 iterationer än det var efter 100 iterationer. Experimenten är utförda med 250 iterationer och det är därför inte möjligt att jämföra eventuella resultat efter till exempel 500 eller 1000 iterationer. Figur 4.6 ger ett exempel på hur en alltför liten ensemble inte har förmågan att få parametervärdena att konvergera till de optimala. Trots att algoritmen körs 900 steg så har parametervärdena inte konvergerat. Det här visar alltså på att experiment 2 begränsas av det icke-optimala utgångsläget.



Figur 4.9: Värdet på kostnadsfunktionen som funktion av antalet iterationer för både experimentens minsta möjliga ensemblestorlek. $F(x)$ motsvarar en ensemblemedlems kostnadsfunktion och μ är här ensembles medelvärde på kostnadsfunktionen. Figur (a) visar experiment 1 med fyra medlemmar och (b) visar experiment 2 med tio medlemmar. Baserat på de här diagrammen ser vi att kostnadsfunktionens värde minskar allteftersom parametervärdena närmar sig det optimala, vilket också är att vänta då parametervärdena förbättras.

Baserat på de här experimenten så framgår att (1) experimentet med större osäkerhet krävde fyra medlemmar för konvergens och (2) experimentet med mindre osäkerhet krävde tio medlemmar för konvergens. Figur 4.9 visar kostnadsfunktionens utveckling som funktion av iterationer för experiment 1 med fyra ensemblemedlemmar och för experiment 2 med tio ensemblemedlemmar. I båda fallen minskar värdet på kostnadsfunktionen och således förbättras parametervärdena gradvis. Ifall förhållandena är optimala, vilket vi har ifall god kunskap om parametervärdena och deras distribution eller osäkerhet finns tillgänglig, är det möjligt att minska ensemblestorleken till ett värde nära parameterantalet. Ifall kunskapen om parametrarna och deras distribution är otillräcklig så begränsar det möjligheterna att minska ensemblestorleken.

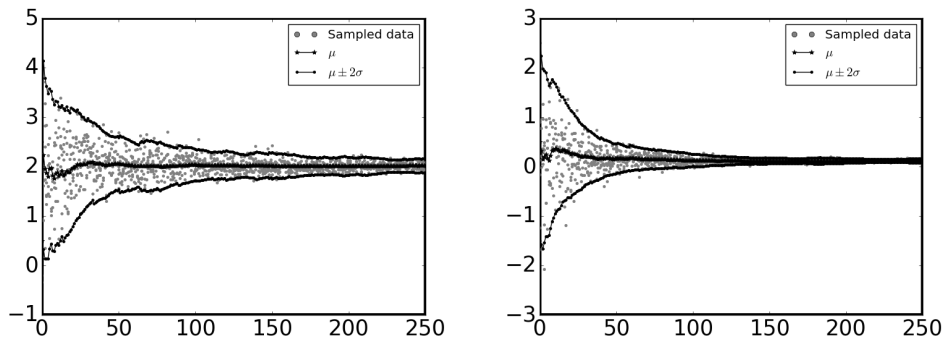
Hypotesen som framfördes i kapitel 1 kan verifieras med hjälp av studier med Lorenz96-modellen. Experimenten visar att algoritmen EPPES inte begränsas av en liten ensemblestorlek utan det är möjligt att optimera två parametrar med en

betydligt mindre ensemble än vad som tidigare har använts. Vid icke-optimala förhållanden, såsom i experiment 2, behövs en ensemble vars storleksklass är större än parameterantalets storleksklass. I experiment 1 är förhållanden optimala och ensemblens storleksklass motsvarar parameterantalets storleksklass. Därmed kan hypotesen verifieras med tillägget att den enbart gäller vid optimala förhållanden.

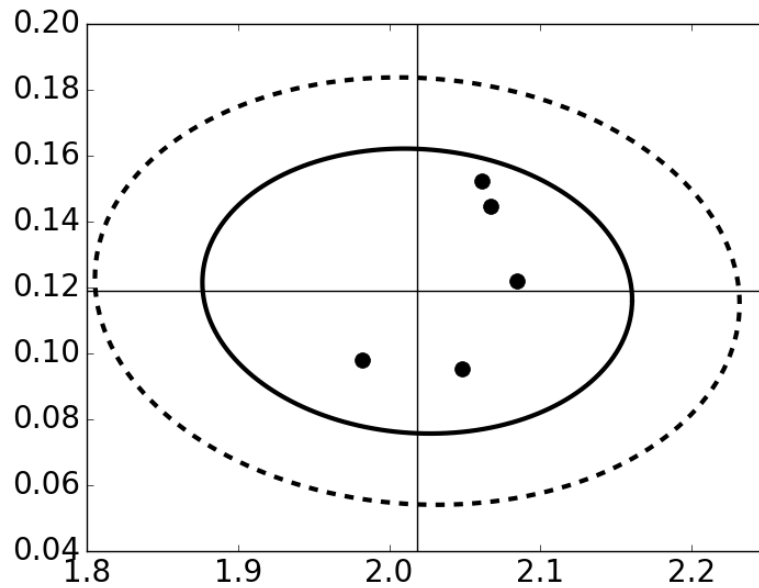
Experimenten med liten ensemble visar på att det finns fler än en faktor som påverkar konvergensen; förutom ensemblestorleken inverkar även parametrarnas initialdistribution på konvergensen. Dessutom påverkar slumpen resultatet - beroende på hurdana parametervärden som samplas konvergerar algoritmen olika snabbt. Det är därmed svårt att på basis av enbart en körning per ensemblestorlek och experiment dra några större slutsatser. För att kunna dra fler slutsatser, t.ex. hur många iterationer som krävs för att en ensemble av en viss storlek ska konvergera eller vilken ensemblestorlek som kräver det minsta totalantalet av prognoskörningar (d.v.s minimum av iterationer multiplicerat med ensemblestorlek), krävs fler körningar för respektive experiment.

Figur 4.10 visar ett exempel på hur utvecklingen av parametrarnas distribution kan se ut vid användning av en liten ensemble, i det här fallet en ensemble på fem medlemmar, samt hur de samplade parametrarna förhåller sig till distributionen. Det är även värt att komma ihåg att slumpen påverkar resultatet genom dragningen av parametervärdena från a priori distributionen. För att få ett mer överskådligt resultat bör därför experimenten göras upprepade gånger för varje ensemblestorlek. Då är det även möjligt att exakt bestämma minsta möjliga ensemblestorlek. I det här fallet är det enbart möjligt att få en första inblick i hur liten ensemblestorleken kan vara och resultaten visar att minsta möjliga ensemblestorlek beror på parameterdistributionens utgångsläge.

Vid optimering av parametrar i en reell väderprognosmodell är det här en viktig faktor att ha i åtanke när experimenten planeras. En reell väderprognosmodell är betydligt mer komplex än Lorenz96-modellen, vilket betyder att resultaten från den här studien inte direkt kan appliceras utan de ger snarare en indikation på vilka sorters resultat som kan förväntas med NWP. För att kunna verifiera hypotesen för en reell väderprognosmodell behövs därför experiment där både parameterantalet och ensemblestorleken varieras. På så vis är det möjligt att få en uppfattning om det finns en relation mellan antalet parametrar och ensemblestorleken för NWP. I enlighet med hypotesen bör förhållandet vara linjärt vid optimala förhållanden. Vid icke-optimala förhållanden så visar den här studien att en något större ensemble behövs.



(a) Parameter 1 med en ensemble på fem medlemmar. (b) Parameter 2 med en ensemble på fem medlemmar.



(c) Distribution efter 250 steg med algoritmen. Notera att initialvärdet på parametern $[0,5; 0,5]$ inte finns innanför 95%- (fylld linje) eller 99%- (streckad linje) sannolikheten. De fyllda cirkelarna (•) motsvarar de parameterpar som har samplats med distributionen och korset visar medelvärde för parametrarna efter 250 iterationer $\mu = (2,02 \ 0,12)$.

Figur 4.10: En ensemble på fem medlemmar ger ett gott resultat vid optimering av två modellparametrar. Figur (a) och (b) visar hur parametrarna konvergerar mot de optimala värdena och figur (c) visar den slutliga distributionen samt de samplade parameterparen. De här diagrammen ger mer tyngd åt verifieringen av hypotesen, då de visar att en ensemble på fem medlemmar räcker för att optimera två parametrar.

Kapitel 5

Sammanfattning

Numeriska väderprognosmodeller är komplexa modeller bestående av många variabler och modellparametrar. En ökad komplexitet av modellerna gör det allt svårare att bestämma parametrarnas optimala värden manuellt. Parametervärdena bör vara så bra som möjligt och oberoende av vädterslag, tidpunkt på dygnet eller årstid. Det här gör det ännu svårare att för hand bestämma deras värden. I stället är det möjligt att använda olika metoder för att optimalt bestämma parametrarnas värden, såsom maximum likelihood, extended Kalman Filter och Markov Chain Monte Carlo-metoder. Algoritmerna har den fördelen att de kan hantera en stor mängd data och därigenom kan de bestämma parametervärden.

I den här studien har en algoritm kallad EPPES, ensembleprognos- och parameteruppskattningssystem (Järvinen *et al.* , 2012; Laine *et al.* , 2012), använts. Algoritmen grundar sig på en Markov Chain Monte Carlo-metod, där idén är att utgående från observationer förbättra a priori distributionen genom att ge mer vikt åt bättre parametervärden, så kallad viktbaserad sampling. A priori distributionen baserar sig på de kunskaper som finns om parametrarna och används vid initialiseringen av algoritmen.

Algoritmen bygger på att variera parametervärdena i ett ensembleprognossystem. För varje ensemblemedlem samplas ett parametervärde eller en parametervektor från a priori distributionen. Ensembleprognosen jämförs sedan med observationer och utgående från vilka parametrar som presterade bra så uppdateras a priori distributionen enligt ett givet schema. Algoritmen antar att det finns en länk mellan parametervärde och prognosförmåga: ju bättre prognosförmågan är, desto bättre är värdet på parametern.

Tidigare resultat har visat att EPPES ger goda resultat vid optimering av ett litet antal parametrar (två och fyra) med en stor ensemble (50 medlemmar) (Laine *et al.* , 2012; Ollinaho *et al.* , 2013a,b). Det är inte nödvändigtvis praktiskt eller

ens nödvändigt att använda så stor ensemble för att optimera så få parametrar. Målet för den här studien var därför att undersöka huruvida det är möjligt att estimerar ett litet antal parametrar med en mindre ensemble. Hypotesen löd som följer:

Hypotes. *Låt L vara antalet parametrar som önskas optimeras och M vara antalet ensemblemedlemmar. Vid en parameteruppskattning med EPPEs så räcker det med att antalet parametrar och antalet ensemblemedlemmar är i samma storleksklass för att konvergens skall uppnås.*

I det här fallet undersöktes hypotesen med hjälp av Lorenz96-modellen (Lorenz, 1996) - en modell som ofta används för att emulera NWP i ensemblesystem. Lorenz96-modellen är en dynamisk modell, som har en kaotisk karaktär vid lämpliga värden på forceringen och modellen lämpar sig därför till att emulera vädermodeller. Modellen kräver minimala beräkningsresurser, vilket möjliggjorde att experimenten kunde utföras med en bärbar dator. Undersökningar med L96-modellen indikerar på hurdana resultat som är att vänta vid användning av NWP.

De numeriska experimenten utfördes med två olika varianter av Lorenz96-modellen. Den fullständiga modellen med både långsamma och snabba variabler användes för att generera syntetiska observationer och den parametriserade versionen, där effekten av de snabba variablerna parametriserats, användes som prognosmodell. I den här studien innehöll den parametriserade versionen en linjär parametrisering: $\theta_1 + \theta_2 X_k$. EPPEs strävade således till att hitta så optimala värden som möjligt för θ_1 och θ_2 så att prognoserna så bra som möjligt motsvarar observationerna.

Här undersöktes hur två olika initialvärden på a priori distributionen påverkas av en minskad ensemblestorlek. I det första experimentet hade a priori distributionen en större osäkerhet och i det andra experimentet hade a priori osäkerheten mindre osäkerhet. Då medelvärdet för distributionen var relativt långt från det optimala värdet, så innebar det här att experiment 1 hade ett bättre utgångsläge än experiment 2 då det optimala värdet låg utanför distributionen i experiment 2, men innanför i experiment 1. I övrigt fanns det ingenting som skiljde de två experimenten från varandra. Ensemblestorleken varierades mellan två och 50 i båda experimenten.

I det första experimentet var det möjligt att minska ensemblestorleken till fyra. I det här fallet var a priori distributionen bred och innehöll de optimala värdena på parametrarna. Efter 250 iterationer hade alla ensembler på fyra eller större konvergerat till eller var väldigt nära det optimala värdet. I det andra

experimenten var det möjligt att minska ensemblestorleken till tio. Orsaken till varför det inte var möjligt att minska ensemblestorleken ytterligare var det icke-optimala utgångsläget. Då a priori distributionen i experiment 2 var snäv så fanns det optimala värdet utanför a priori distributionen och för att nå fram till optimala värdet krävdes det därför fler samplade värden. Således kunde inte heller ensemblestorleken minskas lika mycket.

På basis av de här resultaten kan hypotesen verifieras, med tillägget att den enbart gäller vid optimala förhållanden. Experiment 1 visade att det var möjligt att minska ensemblestorleken till fyra, vilket kan anses vara i samma storleksklass som parametrarna då antalet parametrar var två. I det andra experimenten krävdes en ensemble på tio medlemmar, vilket inte är i samma storleksklass som parametrarnas antal. Trots att det i experiment 2 inte var möjligt att minska ensemblestorleken lika mycket som i experiment 1, så kunde ensemblestorleken ändå minskas till en femtedel av vad som tidigare har använts. Det här visar att det är möjligt att använda sig av en avsevärt mindre ensemble vid optimering av parametrar i Lorenz96-modellen med algoritmen EPPES. Eftersom slumpen har en roll i dragningen av parametervärdena så kan det vara möjligt att minska ensemblestorleken ytterligare, men det här kräver att varje experiment för varje ensemblestorlek utförs upprepade gånger.

	Experiment 1	Experiment 2
Ensemblestorlek	4	10
Storleksklass	10^0	10^1

Tabell 5.1: Sammanfattning av hur många ensemblemedlemmar som behövs för konvergens av två parametrar för de två olika experimenten. Parametrarnas antal är två, vilket motsvarar storleksklass 10^0 . Det betyder att vid optimala förhållanden (experiment 1) så är det möjligt att minska ensemblestorleken så att den motsvarar parameterantalets storleksklass. Vid icke-optimala förhållanden (experiment 2) krävs en större ensemble.

Den här studien har undersökt parameteruppskattningsalgoritmen EPPES och dess förmåga att optimera parametervärden i ett ensemblesystem med en minskad ensemblestorlek. Experimenten med Lorenz96-modellen visar att ensemblestorleken kan minskas så att dess storleksklass motsvarar parametrarnas antal då förhållandena är optimala, d.v.s. då kunskapen om parametrarnas distribution är god. Vid icke-optimala förhållanden är det möjligt att minska ensemblestorleken, men inte lika mycket. De här experimenten indikerar på att motsvarande resultat är att vänta vid optimering av parametrar i en verklig numerisk väderprognosmodell. Det här kräver dock fler och mer omfattande experiment för att kunna bekräfta, vilket är utanför den här studiens ramar.

Litteraturförteckning

- Anthes, Richard A. 1977. A cumulus parameterization scheme utilizing a one-dimensional cloud model. *Monthly weather review*, **105**(3), 270–286.
- Chorin, Alexandre Joel. 1968. Numerical solution of the navier-stokes equations. *Mathematics of computation*, **22**(104), 745–762.
- Gilks, Walter R, Richardson, Sylvia, & Spiegelhalter, David. 1995. *Markov chain monte carlo in practice*. CRC press.
- Hovmöller, Ernest. 1949. The trough-and-ridge diagram. *Tellus*, **1**(2), 62–66.
- Järvinen, Heikki, Laine, Marko, Solonen, Antti, & Haario, Heikki. 2012. Ensemble prediction and parameter estimation system: the concept. *Quarterly journal of the royal meteorological society*, **138**(663), 281–288.
- Kain, John S. 2004. The kain–fritsch convective parameterization: an update. *Journal of applied meteorology*, **43**(1), 170–181.
- Karimi, A, & Paul, Mark R. 2010. Extensive chaos in the lorenz-96 model. *Chaos: An interdisciplinary journal of nonlinear science*, **20**(4), 043105.
- Laine, Marko, Solonen, Antti, Haario, Heikki, & Järvinen, Heikki. 2012. Ensemble prediction and parameter estimation system: the method. *Quarterly journal of the royal meteorological society*, **138**(663), 289–297.
- Lambert, John D. 1973. *Computational methods in ordinary differential equations*. Wiley.
- Leutbecher, Martin, & Palmer, Tim N. 2008. Ensemble forecasting. *Journal of computational physics*, **227**(7), 3515–3539.
- Lewis, John M. 2005. Roots of ensemble forecasting. *Monthly weather review*, **133**(7), 1865–1885.

- Lorenz, Edward N. 1963. Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, **20**(2), 130–141.
- Lorenz, Edward N. 1969. The predictability of a flow which possesses many scales of motion. *Tellus*, **21**(3), 289–307.
- Lorenz, Edward N. 1996. Predictability: A problem partly solved. *In: Proc. seminar on predictability*, vol. 1.
- Lynch, Peter. 2008. The origins of computer weather prediction and climate modeling. *Journal of computational physics*, **227**(7), 3431–3444.
- Navon, IM. 1998. Practical and theoretical aspects of adjoint parameter estimation and identifiability in meteorology and oceanography. *Dynamics of atmospheres and oceans*, **27**(1-4), 55–79.
- Ollinaho, P, Bechtold, P, Leutbecher, M, Laine, M, Solonen, A, Haario, H, & Järvinen, Heikki. 2013a. Parameter variations in prediction skill optimization at ecmwf. *Nonlinear processes in geophysics*, **20**(6), 1001–1010.
- Ollinaho, Pirkka, Laine, Marko, Solonen, Antti, Haario, Heikki, & Järvinen, Heikki. 2013b. Nwp model forecast skill optimization via closure parameter variations. *Quarterly journal of the royal meteorological society*, **139**(675), 1520–1532.
- Palmer, Tim N. 2001. A nonlinear dynamical perspective on model error: A proposal for non-local stochastic-dynamic parametrization in weather and climate prediction models. *Quarterly journal of the royal meteorological society*, **127**(572), 279–304.
- Slingo, Julia, & Palmer, Tim. 2011. Uncertainty in weather and climate prediction. *Phil. trans. r. soc. a*, **369**(1956), 4751–4767.
- Wilks, Daniel S. 2005. Effects of stochastic parametrizations in the lorenz’96 system. *Quarterly journal of the royal meteorological society*, **131**(606), 389–407.
- Wishart, John. 1928. The generalised product moment distribution in samples from a normal multivariate population. *Biometrika*, 32–52.

Bilagor

A Lista över parametrar och variabler

- μ väntevärde, hyperparameter
- Σ kovariansmatris, hyperparameter
- n hyperparameter i EPPES
- W hyperparameter i EPPES
- θ parametervektor
- X långsam variabel
- U snabb variabel
- J antalet snabba variabler per långsam variabel
- K antalet långsamma variabler
- L antalet parametrar
- M antalet ensemblemedlemmar
- p/P sannolikhet
- $y/Y/X_{obs}$ observation
- $N(\mu, \Sigma)$ normalfördelningsfunktion
- $iWish(\cdot, \cdot)$ inversa Wishart-distributionen
- F, F_u forceringstermer
- b, c, h parametrar i L96